FACULTY OF ELECTRICAL ENGINEERING
UNIVERSITY TEKNIKAL MALAYSIA MELAKA

FINAL YEAR PROJECT II REPORT
BEKU 4894

DESIGN AND DEVELOPMENT OF WIRELESS ROBOTIC HAND

Mohamad Fauzi Bin Darmo

Bachelor of Mechatronics Engineering

June, 2014

"I hereby declare that I have read through this report entitle "Design and Development of Wireless Robotic Hand" and found that it has comply the partial fulfillment for awarding the degree of Bachelor of Mechatronics Engineering"

Signature : ……………………………………………………

Supervisor's Name : Engr. Mohd Rusdy Bin Yaacob

Date : 17 June 2014

DESIGN AND DEVELOPMENT OF WIRELESS ROBOTIC HAND

MOHAMAD FAUZI BIN DARMO

A report submitted in partial fulfillment of the requirement for the degree of Bachelor of Mechatronics Engineering

Faculty of Electrical Engineering

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2014

I declare that this report entitle “Design and Development of Wireless Robotic Hand” is the result of my own research except as cited in the references. The report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature :…………………………………………………….

Name : Mohamad Fauzi Bin Darmo

Date : 17 June 2014

Special dedication to my beloved father, mother and my siblings that always giving support in morale and physical.

ACKNOWLEDGEMENT

There are many individual and organization that gives help and support in this project. So, I would like to this opportunity to express my sincere appreciation to all those who provided me the possibility to complete this project and also the report. A special gratitude I give to my final year project supervisor, Engr Mohd Rusdy Bin Yaacob, that assisted me in giving suggestions and encouragement to coordinate my project especially in writing this report.

I am greatly indebted to Shaifulnisam Bin Miskam, the owner of Iqtisam Technology because of his work that done the fabrication for my project. Not only that, he also gives useful advised and guidance especially in the project design.

I also would like to express my gratitude towards my parents and friends which gives encouragement and assistance to me in completion of this project. I also have to appreciate the guidance given by other supervisor as well as the panels especially in my report writing that has improved my writing skill and thanks to their comment and advice.

ABSTRACT

Hazardous work environments such as in radioactive site and space station give high risk to the worker safety. This risk can be reduced by using robot technology that able to take over the human works. For industry purpose, the robot that needs to be developed must have high performance especially in accuracy, stability and consistency. However, the current industrial robot is not flexible which can't be used for the different tasks. So, multi-purpose robot is proposed to be developed in this project. The objective of this project is to design and developed a robot hand with wireless communication. The robotic hand communicates with human in master-slave manner. This manner will help the robot perform for different task without need to reprogram it. In this project, Xbee wireless module is used because it provides a short-range communication channel. The master part is operated by using a data glove that mounted with sensors to measure the flexion of the fingers, which can wear by a human operator. The movement of human fingers will transferred to the slave robotic hand that will mimic the finger movement. The robotic hand prototype has designed by using SolidWorks software. It designed with 5 fingers and has dimensions similar to human hand. Each robotic finger is powered by servo motor and drive by using wire as the actuator. Wire is used to make a link between finger and motor in order to control the finger flexion by rotating the motor. To control the robot hand system, Atmega328 microcontroller is used to for data glove system and also controls the actuation system for robot hand motion. In master part, a microcontroller is used to process the input signal that it measure from the sensor and transmit the signal to the slave robot hand. While in robot hand, a microcontroller is used to process the receive signal and produce output to perform motor operation.

ABSTRAK

Persekitaran kerja yang berbahaya seperti di tapak radioaktif dan stesen angkasa memberi risiko yang tinggi kepada keselamatan pekerja. Risiko ini boleh dikurangkan dengan menggunakan teknologi robot yang mampu untuk mengambil alih kerja-kerja manusia. Robot yang perlu yang akan digunakan dalam sektor industri perlu mempunyai prestasi tinggi terutamanya dari segi ketepatan, kestabilan dan konsisten. Walau bagaimanapun, kebnyakan robot industri pada semasa ini tidak fleksibel dimana ianya tidak boleh digunakan untuk tugasan yang berbeza. Jadi , robot pelbagai guna telah dicadangkan untuk dibangunkan dalam projek ini. Objektif projek ini adalah untuk mereka bentuk dan membangunkan satu tangan robot yang berhubung tanpa wayar. Tangan robot akan berkomunikasi dengan manusia dengan cara tuan- hamba. Cara ini akan membolehkan robot melakukan tugas yang berbeza tanpa perlu diprogramkan semula. Dalam projek ini, XBee modul digunakan kerana ia memberikan saluran komunikasi pada jarak dekat . Bahagian induk "Tuan" beroperasi dengan menggunakan sarung tangan data yang dipasang dengan sensor untuk mengukur kadar lengkungan jari, yang boleh dimakai oleh manusia sebagai pengendali. Pergerakan jari manusia akan dipindahkan ke tangan robot yang akhirnya akan meniru pergerakan jari itu. Prototaip tangan robotik telah direka dengan menggunakan perisian Solidworks. Ia direka dengan 5 jari dan mempunyai dimensi yang sama dengan tangan manusia. Setiap jari robot dikuasakan oleh motor servo dan dihubungkan menggunakan wayar. Wayar digunakan untuk membuat pautan antara jari dan motor yang bertujuan untuk mengawal lengkungan jari semasa motor berputar. Untuk mengawal sistem robot, Atmega328 mikropengawal digunakan untuk untuk sistem di sarung tangan dan juga mengawal sistem penggerak untuk pergerakan robot. Pada bahagian induk , mikropengawal digunakan untuk memproses isyarat dari sensor yang mengukur lengkungan jari dan menghantar isyarat kepada tangan hamba robot. Manakala, mikropengawal di robot digunakan untuk memproses isyarat yang diterima dan menghasilkan output untuk menggerakkan motor.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

The introduction part will explain the project background, problem statement, objectives, scope of research, and contribution of research.

1.1 Project background / Motivation

Robot hand is the machine mechanisms that can mimic the human hand motion in their operation. Grasping stability and fine manipulation are the important factor in the development of the multi fingered robotic hand. In general, the robotic hand is playing the important role in the application that requires precision and dexterity [1]. Besides, the multi fingered robotic hand is also useful to do a work that unreachable by human such as work in hazardous environments, underwater or in a space station. In this project, the prototype of the robot hand is designed by referring human hand. It has designed with 5 fingers and the robot hand has a dimension likely as a human hand. SolidWorks software is used in designing the robot hand.

Data glove is one of the electromechanical devices that used as the controller for robotic hand movement. Large amount of data can be generated from this multisensory data glove device. The controller has some main components which are the microcontroller, wireless module, and sensor. The sensors are mounted on the glove to measure real time human finger flexion. The sensors will be functioning like the potentiometer where it is able to send an input signal for every angle of finger flexion. To achieve high capability in measuring human finger flexion, the sensor that used together with the data glove must be high accuracy and high reliability [2]. The selection of the sensor is made in the literature review section. The electrical system of this controller is designed by using Fritzing software.

Wireless communication is the medium for controlling the robot over a distance where human and robot collaborates to perform tasks [3]. The wireless module must be able to transmit the input signal from the data glove to the robot hand system with fast transfer rate ability. In this project, short-range wireless communication is used because the robotic hand is developed for indoor use only with expected distance around 5 to 10 meters. Wireless module that has a simple dialogue interface and easy to use together with microcontroller may have some advantages to be used in this project. The suitable wireless protocol that meets the criteria needed has been discussed and selected in the literature review section.

1.2 Problem statement

Nowadays, robot is commonly used in the industry. Used of the robot give have high implication especially on the production line. This because, robot able to work with faster and accurate than human. Furthermore, robot is very useful to do a work that unreachable by human such as work in hazardous environments, underwater or in a space station. However, current robot is not flexible which can't be used for the different tasks and the robot needs to be reprogrammed for the different task. In this project, robotic hand is proposed to be design and developed with the manual actuation of a passive system. Manual actuation can ensure that the robot able to operate for different tasks without need to reprogram. The challenging thing in developing multi finger robotic hand is to get the accurate and precise grasp motion. The most important aspects to be considered are the percentage error of the robotic finger bending, stability and reliability. Besides, the robot mechanism must have constant performance in order to achieve accuracy in lifetime measurement.

1.3 Objectives

i. To design and fabricate the prototype robotic hand system together with the electrical circuit.
ii. To develop the master data glove and the actuator system for the robotic finger motion.
iii. To analyze the robotic hand performance in term of wireless communication performance, spring performance, finger accuracy, reliability and consistency.

1.4 Scope

The robotic hand is developed based on the master-slave configuration. Data glove is developed as the master that can control the slave robot hand movement via wireless communication. The robot hand prototype has only five Degree of Freedom (DOF) which is one DOF for each finger. The robot hand will use wire-driven method for the actuation system. The robotic hand system is developed to use for indoor application with a low range wireless communication up to 5 to 10 meters. The robotic hand must be able to mimic human finger motion.

CHAPTER 2

LITERATURE REVIEW

The development of this project includes the designing the data glove and the robotic hand finger. The master-slave system is used in the robotic operation where the robotic finger will mimic human finger motion. This project will use Arduino microcontroller rather than usual PIC. This because Arduino provides a complete, flexible, easy-to-use hardware and software platform while the other PIC microcontroller require to select the type of board circuit, language and also the compiler [5]. The robot hand is designed with 26 Degree of Freedom (DOF). Each finger has 4 DOF which is 1 DOF for adduction / abduction and the other 3 DOF is for extension / flexion [5]. All fingers have a different dexterity level from each other although they have same appearances and shape. This because every finger has different movement ranges with different degree of flexion. The robotic hand applications are very ideal because it can provide precise and stable grasping ability [6]. Tension spring is used to move the finger retract back to initial position at 0 degrees when input signal is low [6]. The spring is attached on every finger joint.

Data glove is the master device that able to control the operation of the robotic hand. The data glove is designed with a sensor that attached on every finger [7]. The sensor is used to measure the angle of finger flexion. Microcontroller is used to measure the input signal from the sensor and sent it to the slave that is the robot hand.

This section reviews the research on the important elements in the development of wireless robotic hand. Since the robotic hand using wireless communication, it is important to do some research on it. The first research study focuses on the protocol for the wireless communication. The second research study discusses the existing of robotic hand and mainly focused on the actuator for finger movement. The third research study discusses the sensor selection for the data glove.

2.1 Wireless communication protocol

Nowadays, a robot that controlled wirelessly is very common. There are a lot of on- going and completed wireless robot project around the world [8]. This robotic hand project is using wireless to make communication between 2 points: robot hand and data glove.

2.1.1 Wireless LAN

Wireless LAN is used as a medium to transfer signal from data glove to robot hand. For the communication module, ultra low power chips are used for the transmitter and receiver hardware that based on a frequency of 2.4 GHz [7]. It is to make sure the communication is possible to make parallel data processing. Based on the wireless network architecture, serial peripheral interface (SPI) is used for communication between transmitter and receiver chip [7]. Multiple streams are sent by multiple transmitted antennas using a transmitter system that have multiple inputs and multiple outputs (MIMO). The stream is transmitted through a matrix channel that consists of multiple paths. The signal vector is received by the receiver and decodes into the original information. Nordic Semiconductor nRF24L01+ transceiver chip is used in the development of the wireless communication. The chip contains transmitter, receiver, 8 bit micro-processor 8051 and 2.4GHz ISM band of frequency operation with GFSK modulation. The data is possible to transfer from 1 to 32 bytes in one data packet since the data transfer rate is settable 250kbps, 1 or 2Mbps. However, this wireless communication has some disadvantage which is in static design. This means new node is not possible to add without reconfiguration during network running.

2.1.2 Bluetooth protocol

Another wireless communication technology that commonly used on the robot is Bluetooth. Bluetooth is the technology with short range Radio Frequency (RF) that used to make point-to-point communication [8]. Bluetooth are highly versatile compared other short range wireless technology because of its feature that is robust against the interferences. To perform a short range wireless communication, class 2 Bluetooth module was chosen with 723kbps maximum data rate. Moreover, class 2 Bluetooth module was chosen compared to the higher class is because of its power consumption [9]. Table 2.1 shows the information about the Bluetooth class. To avoid the buffer overflowing and cancelling the data packets, hardware flow control is actively used with Clear-to-Send/Request-to-Send (CTS/RTS).

Table 2.1: Bluetooth classes [9]

Class	Maximum Power	Operating Range
Class 1	100mW (20dbm)	100 meters
Class 2	2.5mW (4dbm)	10 meters
Class 3	1mW (0dbm)	1 meter

Serial-Port-Profile (SPP) is the communication that is provided by Bluetooth module to interact with each other by using virtual serial port at host [9].

Host Controller Interface (HCI) protocol of Bluetooth module is used to send the data that contains the instruction for the robot to the Handy Board. Bluetooth can perform bi-directional communication where the commands to the HCI will present by the Host (PC) and the Host also receives events back from HCI of Bluetooth module [8]. Bluetooth device is able to support up to 64 kilobytes data packet in length. However, Bluetooth has some limitation where the communication is only one direction and half duplex. Moreover, Bluetooth communication is easily affected by noise.

2.1.3 Zigbee protocol

Zigbee wireless has commonly used for mobile robot as a medium to control the robot movement. Zigbee wireless connection is used to transmit robot motion parameter from PC to robot brain and process the parameters to perform motion control. A Zigbee wireless sensor network is defined as set of low-data-rate communication protocols with short-range wireless networking [10]. The maximum data rate for Zigbee is 250 bits per seconds and it can operate at 868 MHz, 915 MHz, and 2.4 GHz frequency bands. Xbee Zigbee module is used to support Zigbee wireless network and create more stable and reliable network. Mesh networking can be easy and simple to deploy by using Xbee modules. Moreover, Xbee module is easily handled by a microcontroller or a PC because it has a simple dialog interface.

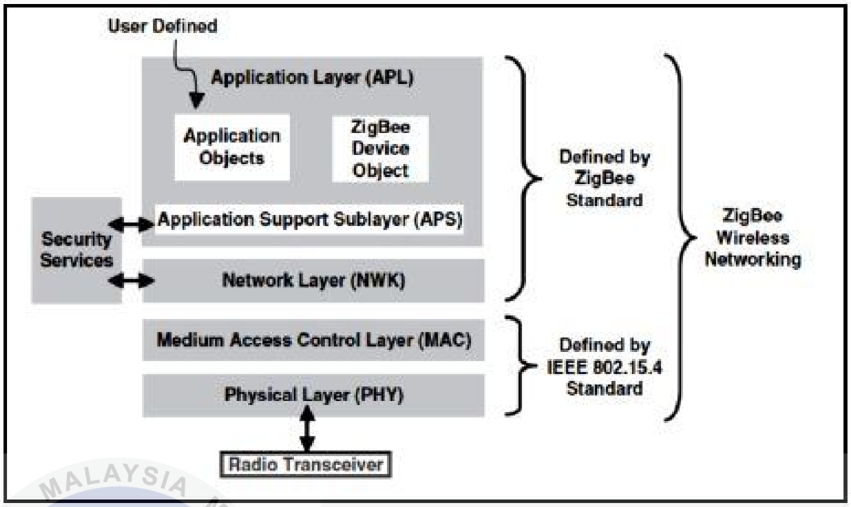


Figure 2.1: Zigbee Wireless Networking Protocol layers [10].

Figure 2.1shows the Zigbee protocol layers which are based on the Open System Interconnect (OSI) basic. Zigbee has some advantages compared to other wireless platform which are more suitable for network application such as larger transmission range, low power consumption and capability for mesh networking [11].

Table 2.2: Wireless technology comparison [11].

	Wireless LAN	Zigbee	Bluetooth
Speed	54Mbps	20~250 Kbps	3Mbps
Power consumption	Highest	Lowest	Medium
Distance	30~70m	30~300m	10~100m
Frequency range	2.4/5Ghz	868 Mhz 915Mhz 2.4Ghz	2.4Ghz
IEEE standard	802.11	802.15.4	802.15.1

2.1.4 Conclusion

In conclusion, Zigbee is the suitable to use as the communication for the robotic hand. This because, Zigbee module has better performance and advantages compared to other wireless platform. Based on the Table 2.2, Zigbee has low power consumption compared to other and it has a better distance range that around 30~300m. For robotic hand application, data speed of 20~250Kbps has enough to meet the requirements for robotic communication.

2.2 Type of actuator

The selection of the actuator that used to drive the robotic hand should be done by considering some critical parameters such as power to weight ratio, efficiency and functionality. This section will discuss the type of actuator that commonly used on robotic hand which is pneumatic actuator, ultrasonic motor and servo motor.

2.2.1 Pneumatic actuator

The purpose if this study is to make some overview of the pneumatic actuator in the development robotic hand. The pneumatic actuator is used in finger joint that have one degree of freedom (DOF). Usually, big compressor is needed when conventional pneumatic actuator is used in order to give enough power to drive a robot hand [12]. The new pneumatic actuator has been developed to get have better performance compared the convenient actuator where consists of rubber balloon and net that covers the rubber balloon.

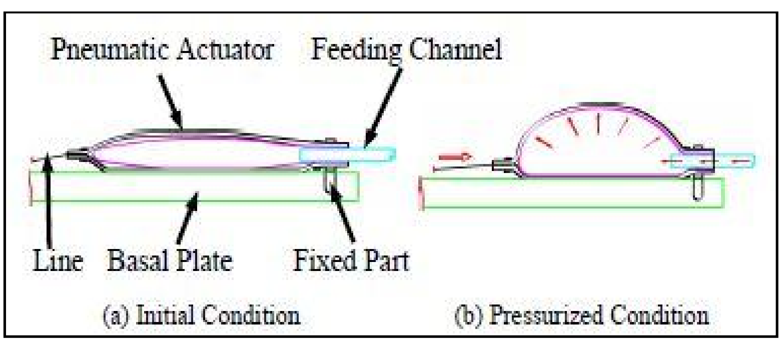


Figure 2.2: Schematic of pneumatic actuator [12].

Figure 2.2 shows a new design of pneumatic actuator with the basic structure that same as the McKibben-type artificial muscle. However this actuator has been improved by covering the net with 0.21mm thickness of rubber balloon that more efficiently to shrink the sent air volume [12]. This can cause the actuator to be driven with low air volume and low pressure. The range operation of robot hand with pneumatic actuator is between -10^o to 90^o and the robotic hand able to grasp and hold objects up to 500g in weight [12].

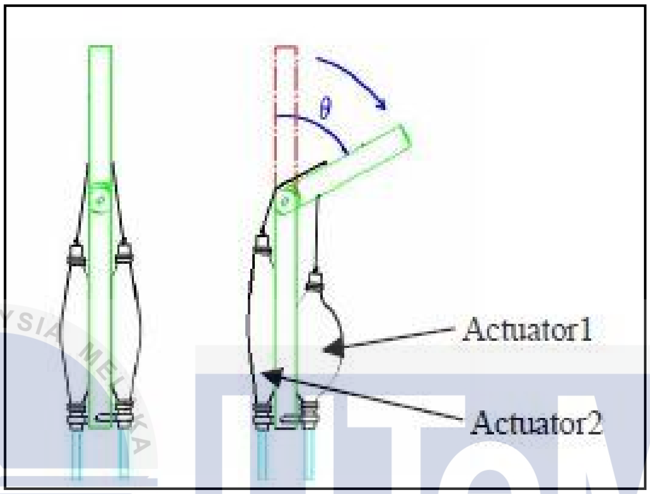


Figure 2.3: Schematic of robot finger with pneumatic actuator [12].

Figure 2.3 shows the operation of the robotic hand finger. When actuators 1 shrink, it will pull the upper part to perform the flexion operation.

2.2.2 Ultrasonic motor

Another actuator that used in the development of robotic hand is ultrasonic motor. This actuator is usually applied to the five-finger robot hand with 20 degrees of freedom (DOF). Ultrasonic is used because it can placed inside the robotic hand since it have characteristic of compact size, light weight and can operate at high torque at low rotational speed [13]. Furthermore, it enables the robot hand to perform a stable grasping motion. However, the robot hand has a risk to break the grasp object since the ultrasonic motor has high driving torque. Moreover, there is some difficulty to control the torque because of the non-linearity characteristic of the ultrasonic motor. Spring is used together with the ultrasonic motor in order to control the elasticity of the joint. By changing the spring coefficient, robot hand would obtain high stability in grasping objects [13].

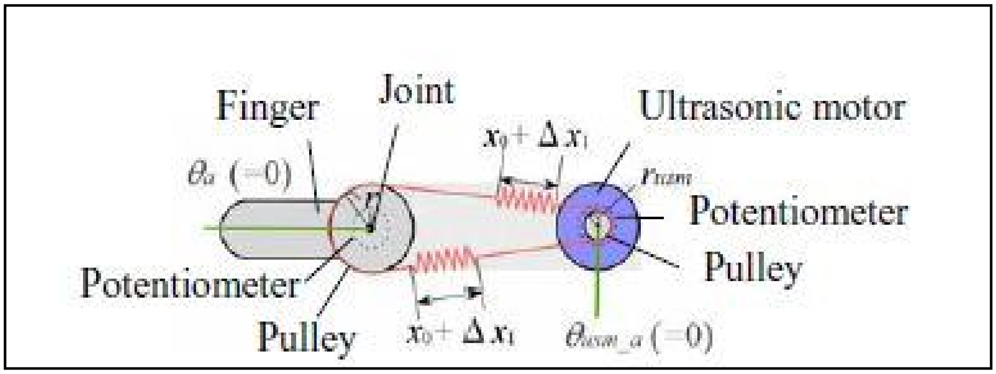


Figure 2.4: Outline of driving method of finger [13].

The rotation angle of the ultrasonic motor is determined by using a potentiometer. The relationship of the motor rotation angle with joint angle can be expressed as:

$$R= \theta_b = r_{usm}.\theta_{usm\_b} \tag{2.1}$$

To achieve highly accurate force control, ultrasonic motor is the suitable actuator because it's precise position control. Besides, the stable and efficient grasping motion can be achieved if the restoring force is used as the output force.

2.2.3 Servo motor

Servo motor also can be used as the actuator for the robotic hand. Servo motor has good performance in speed control and also the torque control [14]. There is also has small size of servo motor that suitable if used in robotic hand. The speed of the finger movement can be adjusted by controlling the rotational speed of the servo motor. PWM control is to adjust and set the servo motor rotational speed. The control period of the servo motor is 0.18us with speed and torque control of 0.125% accuracy error. In order to grasp brittle object, the performance of the servo motor can be tuned by the driver gain to show no overshoot. Figure 2.5 shows the graph of the relationship between finger joint and servo motor. From 'moving' region on the graph shows the joint faster output than input motor speed while the output torque is lower than input motor torque.

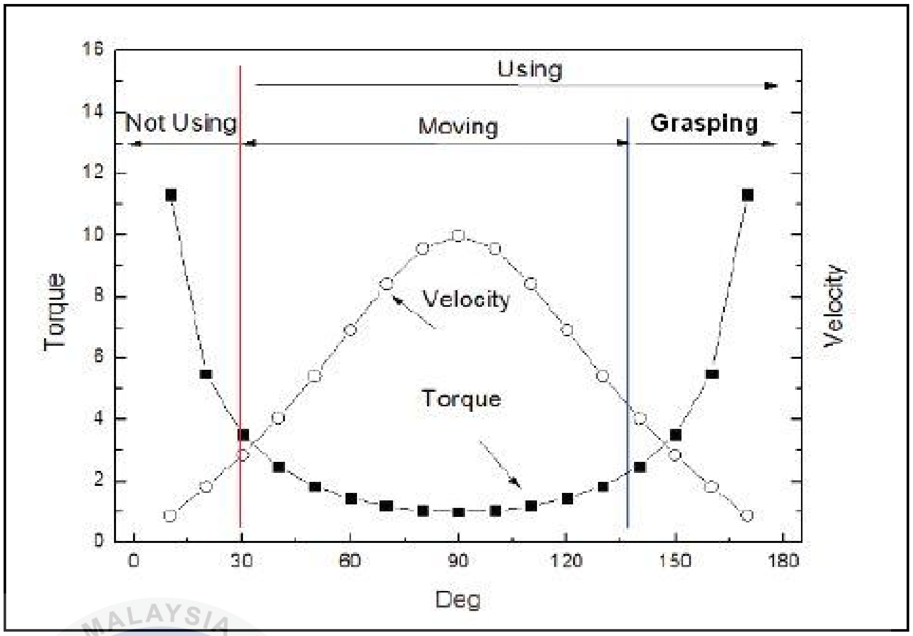


Figure 2.5: Motor rotational speed/torque versus joint speed/torque [14].

2.2.4 Conclusion

As a conclusion, servo motor has better performance if used as the actuator for the robotic hand. This because the speed of the robot finger flexion can be easily controlled by adjusting the speed of the servo motor using PWM. Besides that, servo motor has low voltage of the motor winding that only 12V [14]. Low torque and speed motor is required in order perform the robotic hand motion smoothly especially in grasping ability. Since the project used wire-driven for finger movement, the servo motor is the suitable choice as the actuator.

2.3 Type of sensor

For this robotic hand project, sensor is used in the data glove. It is used to read the human finger flexion as the input signal. In this section, several types of sensor that commonly used in the development of robotic hand is discussed which is tactile sensors, flex sensor and pneumatic force feedback method.

2.3.1 Tactile sensor

The tactile sensor is the one if the sensor that able to detect flexion. To make the fabric tactile sensor, modified artificial hollow fiber is weaving into the cloth (glove). The contact force is detected by the sensor when the warp and weft intersect. The sensor will sequentially scan and measure all capacitances change at all points and obtain the capacitance value [15]. Both warp and weft fiber will deform when load is applied at the intersection point of the fabric tactile sensor. As the load applied, the value of the capacitance is increased and the contact force can be measured from it. The material of artificial hollow fiber is the silicone rubber tube with outer diameter of 250µm and 170µm inner diameter.

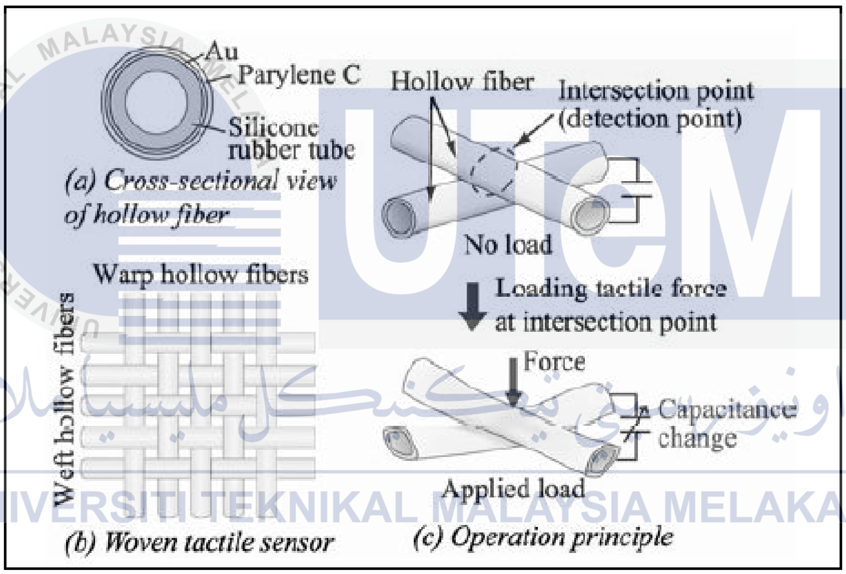


Figure 2.6: Fabric tactile sensor [15].

As shown in Figure 2.6, metal and insulation layers are used to cover the surface of elastic hollow fiber and the sensor is produced by weaving the hollow fiber together as shown in (b). The performance of the sensor output is shown as the output value will increase when the load applied is increased. The variation in the output was less than 10% compared with the initial value due to the change of the tension.

2.3.2 Flex sensor

The commonly used of sensor to measure finger flexion is the flex sensor. It used as an input device for collecting and classify the finger gestures. To track the bending movement of fingers, 4.5inches of flex sensor is used [16]. Flex sensor behaves like analog resistance and operate as an analog voltage divider. The acquire data that relative to the amount of finger bending can be obtained by connecting the electronic system to the sensor [17]. The used of this sensor are suitable in development of data glove because it can minimize the use of wire. With the experiment that has been performed, the flex sensor shows a good performance in term of accuracy and repeatability where it had above 90% of accuracy with 10 samples per gesture [16]. Other than that, the flex sensor able to detect the minimum bend finger with 5% of sensor value.

2.3.3 Pneumatic force feedback

Another method that used to measure finger gesture is by using pneumatic force feedback data glove. This pneumatic force feedback data glove has a simple structure with high force to weight ratio and linear control force [18]. The pneumatic system used micro-low-friction cylinder as the actuator and non-contact magnetic sensor to measure the finger gesture. The parameter that mainly focuses on this pneumatic force feedback is the power-to-weight ratio, mechanical structure, friction and also the force controllability. The displacement of the piston is 30mm in maximum, range of the piston rotation angle between upper and lower is -25.5^o to 40^o while the rotation angle is -25.5^o to 25^o for right and left movement [18]. To calculate the current hand gesture, the non-magnetic contact sensor is used to measure cylinder rotation angle while the piston displacement is measured by non-contact proximity switch. The cylinder has 0.03Mpa to 0.6Mpa of working pressure with maximum force of 47N. The measurement accuracy of the cylinder is 0.05mm with ±0.1mm of repetition accuracy [18].

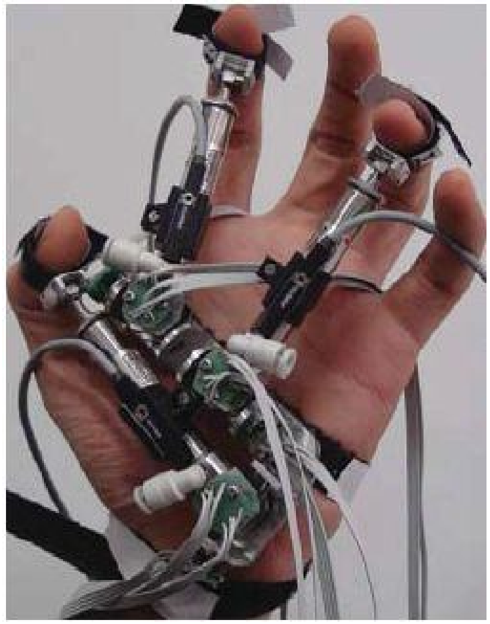


Figure 2.7: Prototype of pneumatic force feedback data glove [18].

2.3.4 Conclusion

In conclusion, pneumatic force feedbacks data glove and flex sensor have better performance in term of accuracy and repeatability. However, pneumatic system has a large mechanism compared to flex sensor. Besides, the complicated mechanism of the pneumatic system causes it hard to develop. The pneumatic system seems not compatible with this robotic hand project because it only supports up to 3 fingers only while the project use all 5 fingers. Flex sensor is the best choice to use in this robotic hand project as well it has a simple mechanism with good performance.

CHAPTER 3

METHODOLOGY

3.1 Introduction

Project methodology describes the list of procedure or methods that performs for the project development. The theoretical analysis of the method applies to this project will describe in this chapter. It consists of process flow chart, development process and experiment conducted.

3.2 Project activity and planning

The project has started to review the concept of the robotic hand. The project review is based on the system operation, software and hardware implementation. Next progress is identification of problem that based on the project development. The problem statement includes the system performance and technical error. Project objective is the main goal that needs to be achieved in this project. Based on the previous research that related to this project, literature review has been written out. The literature review is important because it works as references and help in understanding the project progress, problem and as the resources for analysis. The development process is the process for software and hardware design. After the designed is confirmed, the prototype is fabricated based on the drawing design. The prototype then needs to be tested and troubleshoot the function and performance.

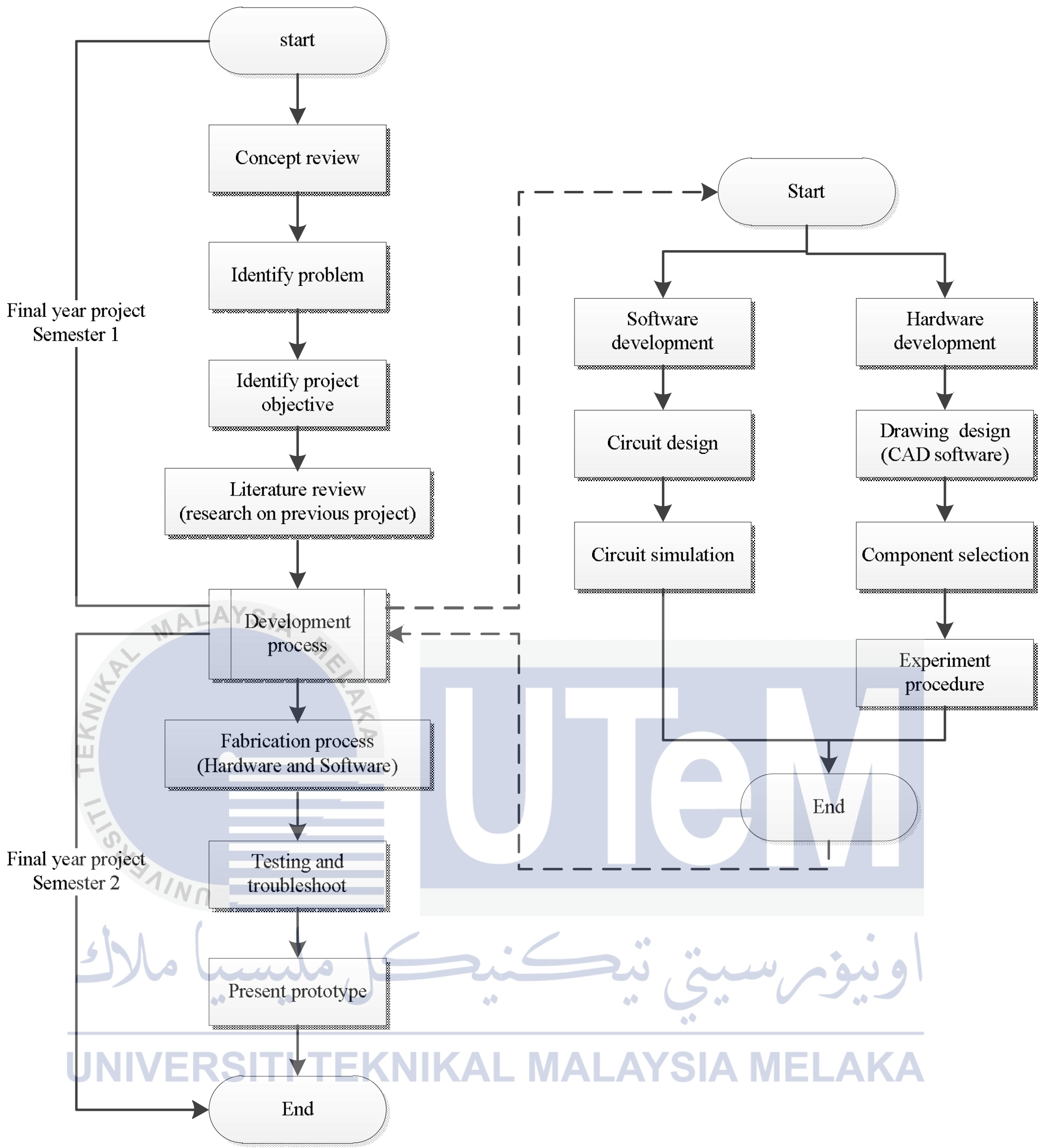


Figure 3.1: Flow chart for whole project.

3.2.1 Milestone

Table 3.1 shows the milestone of activity that has been planned. The activity planned may be changed depending on the situation and advised from supervisor.

Table 3.1: Planning activity.

No	Activity	Date
1	Implement of the project - Theoretical study - Hardware design - System design	September 2013 until November 2013
2	Hardware implementation	November 2013 until January 2014
3	System implementation	December 2013 until February 2014
4	Prepare the required sample for test guideline	February 2014
4	Running an experiment and collect all required data.	February 2014 until March 2014
5	Hardware test and troubleshoot	March 2014 until April 2014
6	Hardware performance analysis	April 2014 until May 2014

3.3 Development process

This section will explain the operation process between the robotic hand and the master data glove. From Figure 3.2, a sensor that attached to the data glove is bent to measure the input signal from human finger flexion. That signal is then transmitted though wireless connection by using the Xbee module. The robot will receive the signal and process it by using Arduino microcontroller and then operate the motor to perform the motion of the robot finger.

3.3.1 Process flow chart

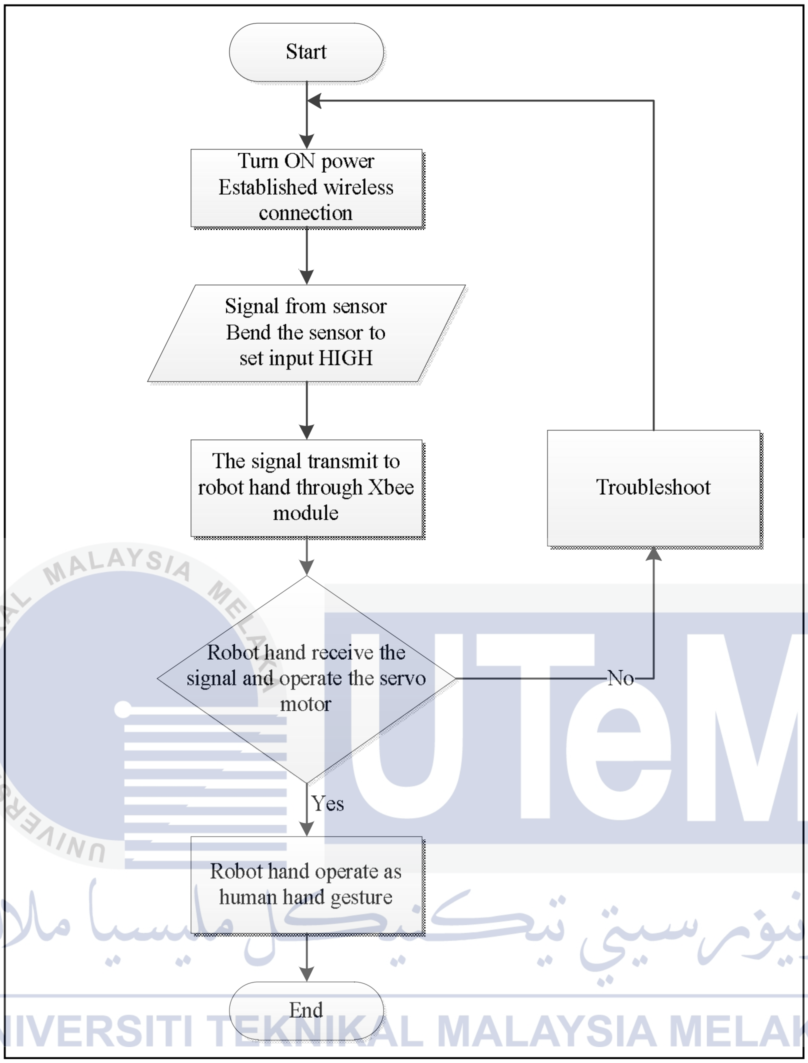


Figure 3.2: Process flow chart

Figure 3.2 shows the process flow chart of the operation of wireless robotic hand. The system is turn ON for both transmitter and receiver system. After the system in turn ON, the communication between data glove and robotic hand will establish through the wireless system. The input signal will obtain after the sensor measure the finger flexion and transmit to the robotic hand through Xbee module. The servo motor that used as the actuator for the robot fingers will operate and the rotation of the servo motor will depend on the input value. Robotic hand will mimic the human finger motion and test the robotic grasp ability by grasping an object.

3.3.2 Mechanical system

The mechanical system has included the development of a prototype of robotic hand and also the actuation system. The robotic hand has been designed by using SolidWorks software. The robot hand is designed with 5 different parts that include hand palm, proximal phalanx, middle phalanx and distal phalanx and the connector. The robot hand is designed to have a dimension that similar to the human hand and the general dimension of robot hand is 323mm height x 178mm length x 22mm width as shown in appendix A. Figure 3.3 shows the hand palm for right hand mechanism. The hand palm is included the holder for each finger. Servo motor that actuates the finger motion will mount on the back of the hand palm. Figure 3.4 is the design for the proximal phalanx, Figure 3.5 is the middle phalanx and Figure 3.6 is the distal phalanx of the robot index finger. All 5 fingers have different dimension although it has the same design. All the finger parts will assemble together with the hand palm by using the connector as shown in Figure 3.7. The other end of the connector wills tight using e-clip.

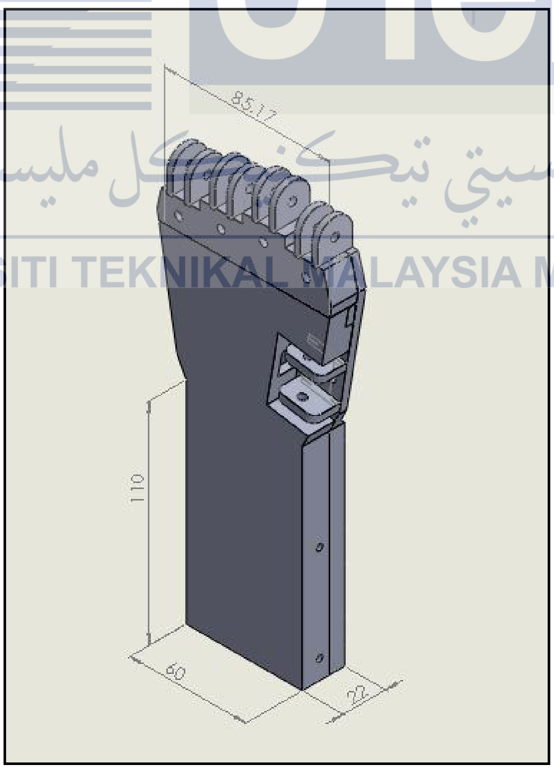


Figure 3.3: Right hand palm

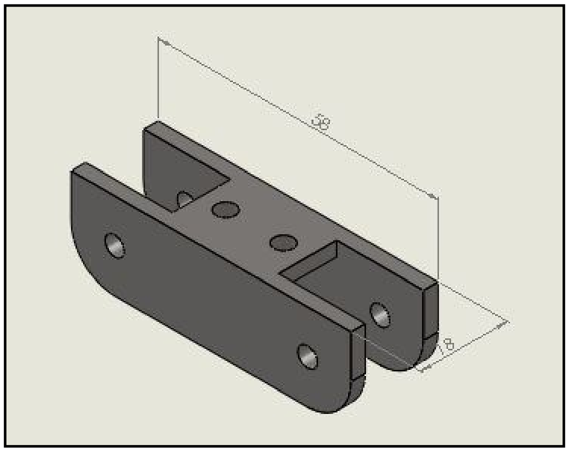


Figure 3.4: Proximal phalanx

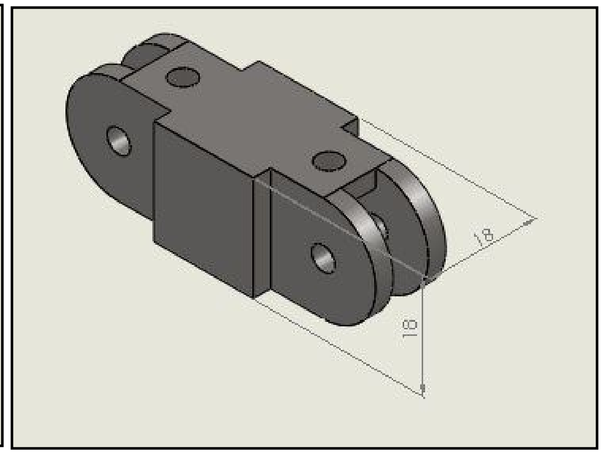


Figure 3.5: Middle phalanx

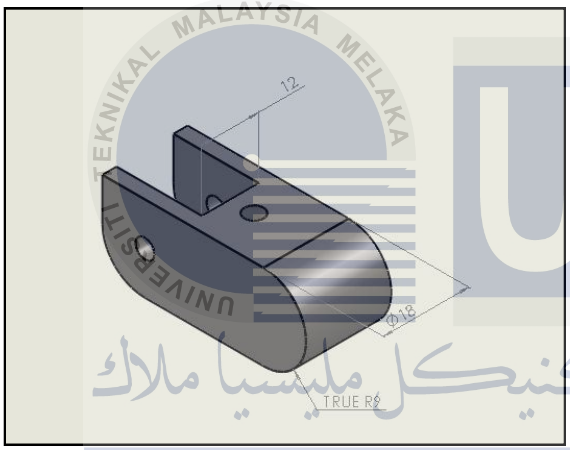


Figure 3.6: Distal phalanx

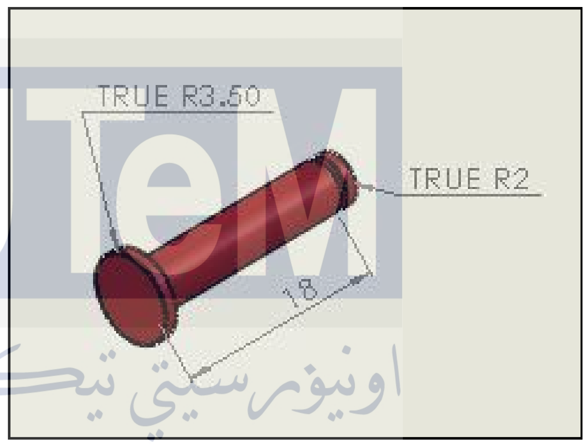


Figure 3.7: Connector

The materials use for the fabrication of the robot hand is Aluminum 6061. The fabrication process by using the aluminum covers all area of the robot hand where the specification for all part is state in the Table 3.2

Table 3.2: General hand specification

No	Parts	Specification			
		Width (mm)	Length (mm)	Height (mm)	Thickness (mm)
1	Palm Base	8.00	90.00	200.00	4.00
2	Palm Cover	14.00	88.33	175.00	4.00
3	Finger Base	3.00	18.00	19.00	3.00
4	Thumb Base	14.00	20.00	4.00	4.00
5	Thumb Base 1	18.00	20.00	22.00	3.00
6	Thumb Base 2	32.00	61.71	30.00	3.00
7	Thumb Top	18.00	18.00	48.00	3.00
8	Thumb Inner	18.00	18.00	53.00	3.00
9	Index Inner	18.00	18.00	58.00	3.00
10	Index Middle	18.00	18.00	53.00	3.00
11	Index Top	18.00	18.00	34.00	3.00
12	Middle Inner	18.00	18.00	58.00	3.00
13	Middle Middle	18.00	18.00	58.00	3.00
14	Middle Top	18.00	18.00	43.00	3.00
15	Ring Inner	18.00	18.00	56.00	3.00
16	Ring Middle	18.00	18.00	53.00	3.00
17	Ring Top	18.00	18.00	38.00	3.00
18	Pinky Inner	18.00	18.00	54.00	3.00
19	Pinky Middle	18.00	18.00	46.00	3.00
20	Pinky Top	18.00	18.00	38.00	3.00
21	Connector	4.00	20.50	4.00	4.00

3.3.3 Electrical system

Two controllers are designed for the robotic hand since the robot used wireless communication. The circuit for both controllers has been designed by using Fritzing software. Fritzing software is used because it can design a circuit with actual electronic component that use in this project. One controller is designed for data glove that used to measure the sensor value and the other controller is to operate the servo motor for finger flexion. The circuit is designed based on the selected electrical component.

Table 3.3: Component used

Data glove	Robot hand
Arduino UNO with Atmega 328 Xbee module 5 flex sensors 5 resistors	Arduino UNO with Atmega 328 Xbee module 5 servo motor

Arduino with Atmage328 microcontroller is used to measure the input signal from a flex sensor that connect to analog pins. Xbee module is used as a transmitter to transmit the input signal to the robot hand. For robot hand controller, Xbee module is used together with Arduino to receive the input signal from the data glove. The microcontroller is used to process the signal and then operates the servo motor. For both circuits, Xbee is connected to the RX and TX pin of Auduino kit.

The electrical system for the master data glove is shown in Figure 3.8. The figure also shows where the position of the flex sensor that will be mounted on the glove. Each of the flex sensors is connected with the resistor is series before it connect to the analog port of the Arduino kit. The Arduino is mounted together with Xbee to performing serial communication.

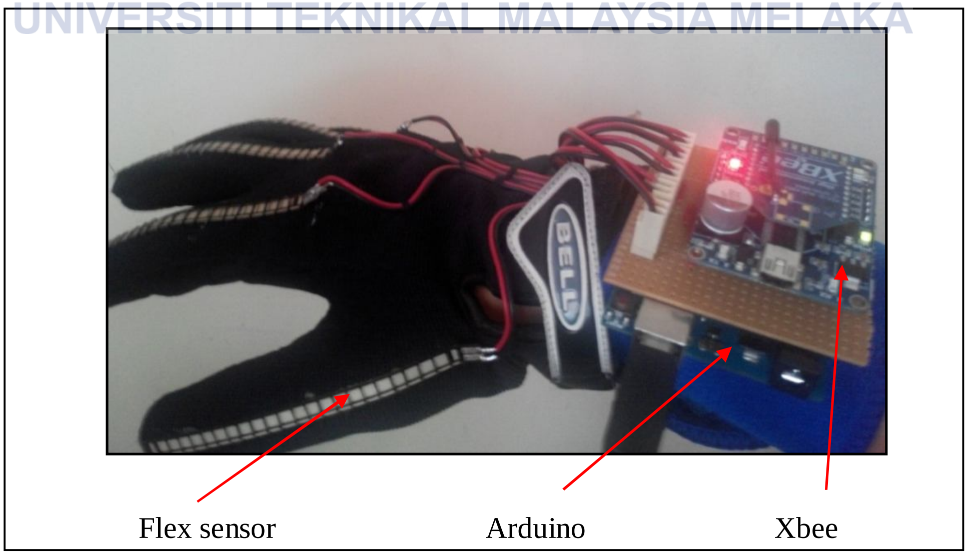


Figure 3.8: Master data glove

3.3.3.1 Xbee module

Xbee module that used in this project is Xbee series 1that also known Xbee 802.15.4. This because Xbee series 1 module use IEEE802.15.4 networking protocol that providing fast peer-to-peer networking or point-to-point/multipoint communication. The advantage of this Xbee module is it supports application that requires low latency and it suitable for low-cost, low-power applications. From the specification of Xbee Series 1 as shown in Table 3.4, this module has meet requirement needed to use in robotic hand project. This because, the robotic hand system is developed for indoor application with a low range wireless communication around to 5 to 10 meters only. To use the Xbee more easily, it is used together with Xbee Explorer USB (Figure 3.9) because the Xbee can be mount into it without soldering. It is the best kit to use especially when to configure the Xbee in X-CTU software.

Table 3.4: Xbee Series 1 specification

Performance	**Xbee Series 1**
Power output	1mW (+0 dBm)
Indoor/Urban range	Up to 100 ft (30 m)
Outdoor/RF line-of-sight range	Up to 300 ft (90 m)
Receiver sensitivity	-92 dBm
RF data rate	250 Kbps
Operating frequency	2.4 GHz
Interface data rate	Up to 115.2 Kbps

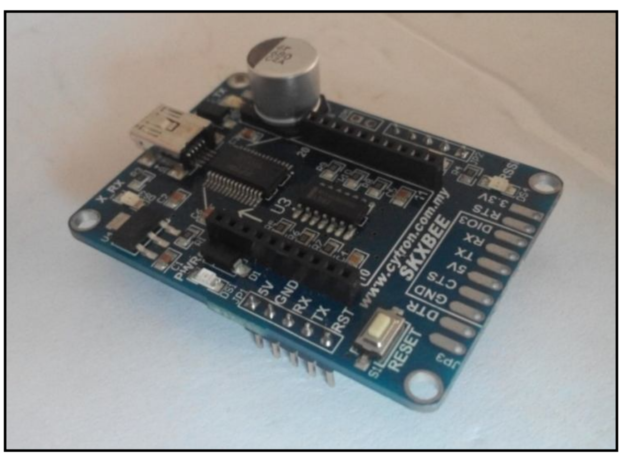


Figure 3.9: Xbee Explorer USB

This project only used to Xbee to communicate and AT mode (transparent mode) is for this point-to-point communication because only two Xbee is used in this project that is the Coordinator (sender) and End Device (receiver). Since the Xbee module is from Series 1, the important setting parameter is Channel, PAN ID, Baud Rate, Destination Address Low (DL) and 16-bit Source Address (MY). Both Xbee setting must have same value for the setting parameter except the DL and MY where it need to set vice versa. The setup for both Xbee is as shown in Table 3.5 below.

Table 3.5: Xbee configuration

PARAMETER	COORDINATOR	END DEVICE
CH-Channel	C	C
PAN-ID	3100	3100
DL-Destination Address Low	1234	5678
MY-16-bit Source Address	5678	1234
Baud Rate	9600	9600
CE-Coordinator Enable	1-COORDINATOR	0-END DEVICE
AP-API Enable	0-API DISABLE	0-API DISABLE

3.3.3.2 Arduino Microcontroller

Microcontroller use for this project is the Arduino Uno R3. It is selected because the Arduino have enough pin to use for this project either the analog or digital pins. Two Arduino is used in this project since the system is not controlled directly by computer (PC). The Arduino is responsible to read input from the sensor ad transmit it through Xbee communication and finally control the actuator. The Arduino programming is quite simple but there is important command in the programming that to perform serial communication with Xbee.

```
void setup()
{
  Serial.begin(9600);
}
```

Figure 3.10: Serial command

The command is as shown in Figure 3.10 where 9600 is the baud rate value. This value must be same as Xbee baud rate. The serial communication cannot be performed if the value is different. This command setup is same for both transmitter and receiver.

In the transmitter system, Arduino is responsible to read the analog input value from the flex sensor. However, each sensor have different range output as example sensor1 has range between (0.68v – 2.35v) while sensor2 has output range between (1.64v – 3.03v). The problem is servo motor require input voltage between 0v to 5v in order to fully functioning where 0v is to put the servo in its minimum position (0°) and 5v to its maximum position (180°). This means the output voltage from the sensor is not enough to supply to the servo. To overcome this problem, mapping function is used in the Arduino programming.

```
int value4 = map(sensor4, 480, 140, 0, 179); // (9)
int value5 = map(sensor5, 300, 620, 0, 179); //(10)
```

Figure 3.11: Arduino mapping function

The map function is map(value, fromLow, fromHigh, toLow, toHigh). By referring Figure 3.11 (value4), 480 is the maximum Bytes and 140 is the minimum Bytes for that sensor. The mapping setup for “value4” is differently if compare to “value5”. This is because the servo that corresponds to it is required to functioning in reverse direction. To use this mapping function, the output voltage from the need to convert into Bytes. This bytes value can be obtain directly from the Arduino without require extra calculation since the Arduino already read the analog value in Bytes. To send the mapping value through the Xbee, the value must be write in the serial as shown in Figure 3.12.

```
Serial.write(value4);
Serial.write(value5);
```

Figure 3.12: Serial write function

In receiver part, Serial.available command is use to make sure the received signal before proceed the entire program. For this project, the receiver command is setup as Figure 3.13 where value “5” means the incoming 5 input from the transmitter. It because of total 5 flex sensors is used in this project. If the incoming serial data is verified, the programmed proceed with read the serial data and control the servo as shown in Figure 3.13.

```
void loop()
{
  if (Serial.available() >= 5)
  {
    //read serial data from input pin
    byte templ = Serial.read();
    Serial.println(templ);
    myservo2.write(templ);
  }
```

Figure 3.13: Serial available

3.3.3.3 Flex Sensor

Flex sensor is the important components that use in this project in order to obtain the angle of the finger flexion. 4.5 inch of flex sensor length is the best choice since the length almost same as finger length. Because of that, the flex sensor will nicely mount on the glove. This condition cause sensor bent exactly in middle area of its length where the best flex sensor reading is on the middle area of the sensor.

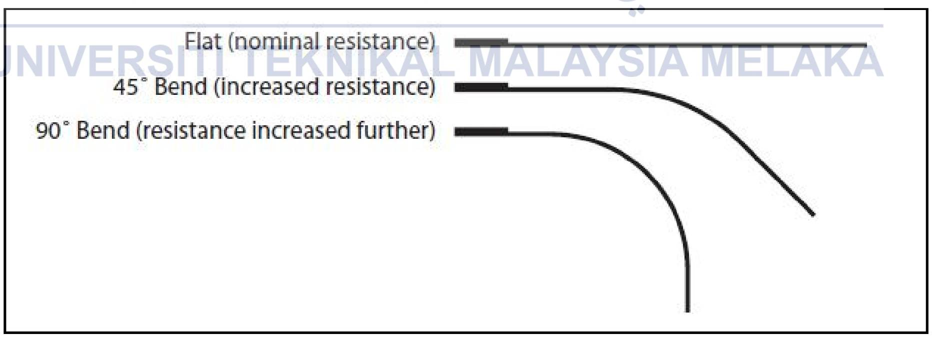


Figure 3.14: How flex sensor works

Figure 3.14 shows the flex sensor working operation where the resistance will increase it bend further than the flat condition. The standard resistance value of the flex sensor is as in Table 3.6 for different bending. The resistance specification is taken from http://cytron.com.my website.

Table 3.6: Standard resistance specification

Bending angle (θ°)	**Resistant (Ω)**
0°, Straight (unflexed)	~9000 Ω
45°	~14000 Ω
90°	~22000Ω

However, after done the testing on each of the flex sensor, all of them do not have the same specification as in Table 3.6. Each of the sensors will provide different output value and also have different range value after the test done in between 0° to 180°. The sensor testing is done by using the Arduino where the result is in Bytes. This result is monitor in real time graph by using Bridge Control Panel software. The example of the result is shown in Figure 3.15 where it is obtained from the flex sensor that mounts on the index finger.

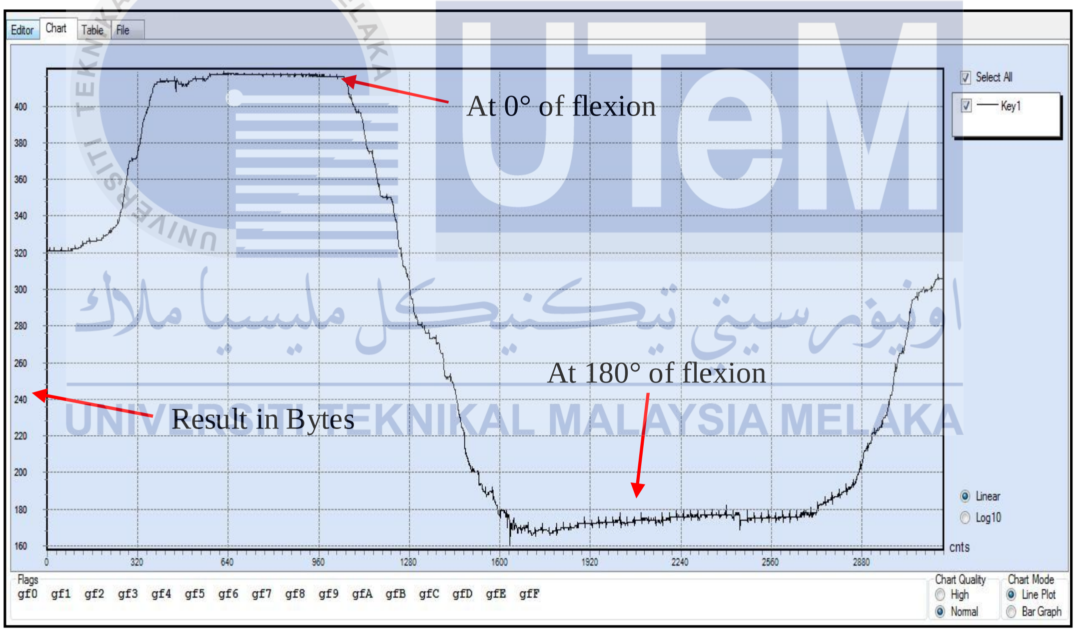


Figure 3.15: Sensor graph pattern by Bridge Control Panel software

The different output value of each the sensor can be prove in the Arduino coding as in Appendix A where the mapping command has different value for every sensor as shown in Figure 3.11.

To obtain the real time graph as in Figure 3.15, some command is needed to make the Bridge Control Panel software synchronize with Arduino and perform communication in serial. The important command is shown in Figure 3.16 where the upper side for Arduino and the below side is for Bridge Control Panel.

```
void loop()
{
  // read the input on analog pin 0:
   int data = analogRead(A0);

   Serial.write(data>>8);
   Serial.write(data&0xff);
}
```

```
RX8 [h=43] @1Key1 @0Key1
```

Figure 3.16: Command to perform serial communication

The important command for Arduino is the Serial command to ensure the Bridge Control Panel can read the data and plot it in the graph. The command for the Bridge Control Panel can be write in the Editor mode. All the sensor output data is recorded in the table as in Chapter 4.

3.3.3.4 Servo Motor

For actuation system, servo motor has been selected to use in this project to drive the finger flexion. The robotic hand requires a total of 5 servo motors where 1 servo motor is used for each finger. In this project, all of the servo motor will placed in the hand palm of the robot hand and wire is used to connect the servo motor with the finger. The RC micro servo motor is used in this project. This because the dimension of robot hand can be minimize by using this micro servo motor. Moreover, the performance of the servo motor is meet the requirement needed in order to drive the robotic fingers. Turnigy TGY-90S micro servo motor is the best servo to use as the actuator as the servo specification is state in Table 3.7.

Table 3.7: TGY-90S micro servo specification

Item	Specification
Operating speed (at no load)	0.10 sec/60^{o} at 4.8V 0.08 sec/60^{o} at 6.0V
Stall torque	1.8 kg-cm at 4.8V 2.2 kg-cm at 6.0V
Limit angle	180^{o} ± 10^{o}
Weight	17 ± 1 grams (without servo horn)

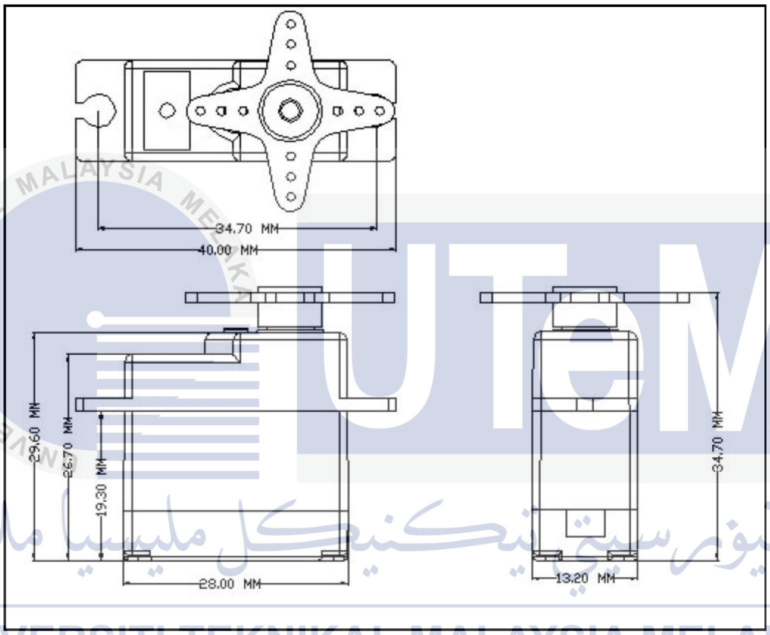


Figure 3.17: Dimension of RC Micro servo motor

Another reason of selecting TGY-90S micro servo is because it used metal gear type where it is suitable to use for heavy duty application. Moreover, metal gears will provide high durability compare to plastic gear because they not easily fracture and wear out. Other advantage of the micro servo is the gear has small teeth. This type of teeth will provide high precision for each rotation angle.

Based from the specification, the servo motor has limit angle of 180° but in the real servo, it can’t reach its limit angle due to the limitation from the potentiometer. The potentiometer is placed inside the servo motor to control the output signal and also control the position the servo rotation. To overcome this problem, the micro servo motor needs to undergo some modification by adding 2.2kΩ resistor to both positive and negative terminal of the potentiometer. Figure 3.18 shows the condition inside the servo motor after the modification.

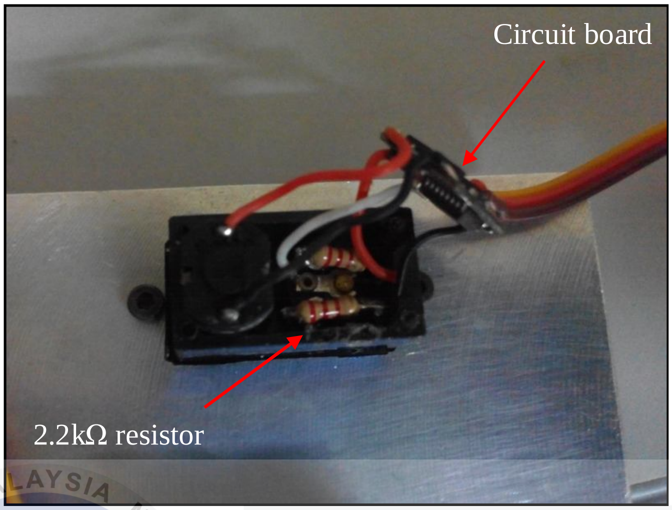


Figure 3.18: Servo motor after adding 2.2kΩ resistor

The servo motor able to performed full rotation from 0° to 180° after the modification. However, there are some risks after the modification where the servo motor can overshoot and caused the gear jammed. This problem can be counter back by adjusting the sensor mapping in the Arduino coding.

3.4 Experiment

This section is about the experiment has been done to test the robotic hand performance. The section consists of 4 type experiment that is sensor performance, spring performance and measurement of finger position.

3.4.1 Wireless communication performance

The objective of this experiment is to measure the performance of Xbee Module in term of time delay for differences mounting elevation over the distance. The second objective is to find the maximum position that includes the distance and elevation for transmitter placement.

In this experiment, the transmitter (coordinator) required to send 32 Bytes of data. The 32 Bytes of data is write is ASCII mode and the data is:

0123456789:;<=>?@ABCDEFGHIJKLMNO = 32 Bytes/transfer

3 different elevation parameters are set for this experiment that is -1.5m, 0m and 1.5m. The elevation is change on the position of the transmitter only and the receiver is placed in fixed position. -1.5 meter of elevation is means that the transmitter is placed 1.5 meter below the receiver and 1.5 meter is above the receiver. The illustration of the experiment is as shown in Figure 3.19.

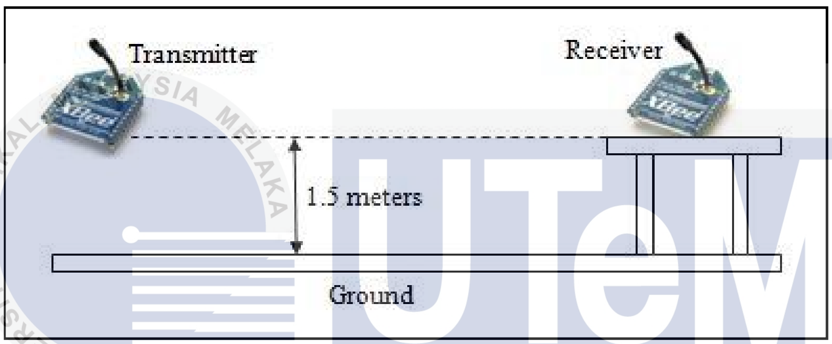


Figure 3.19: Experiment illustration

From the Figure 3.19, the receiver is placed in fixed position above the ground. The experiment parameter that changes is only the position of the transmitter. This condition is selected is because in real operation, the robot will placed a position and the only position that change is the controller that is the master data glove that wearing by human.

Docklight V2.0 software is use in this experiment in order to obtain the results. Docklight is a tool to make simulation and analysis for serial communication protocol. By using this software tool, the data transfer between two Xbee can be monitor. To perform the communication by using Docklight, some configuration need to setup that include selecting the serial port that attach with Xbee such COM6 for the receiver is this experiment. However, the COM value is depending on the PC used. For the data transfer, the data packet is setup is shown in Figure 3.20 and the data is set to send in sequences for each 5 seconds.

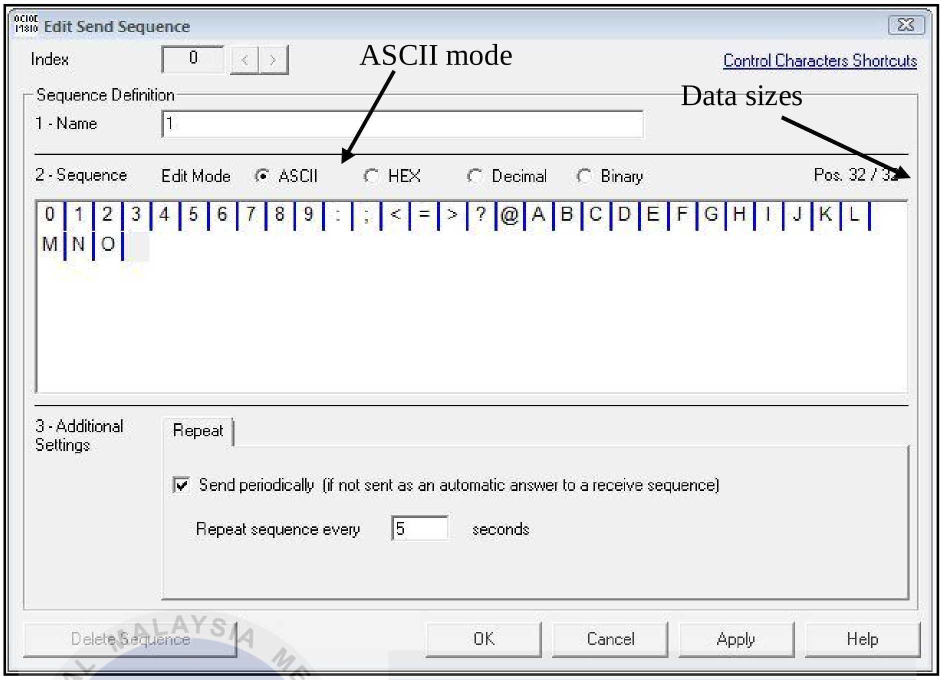


Figure 3.20: Setup data packets

As a result, each incoming data that receive by the software is includes with the time where it can be used to calculate time delay between the transmitter Xbee with the receiver Xbee.

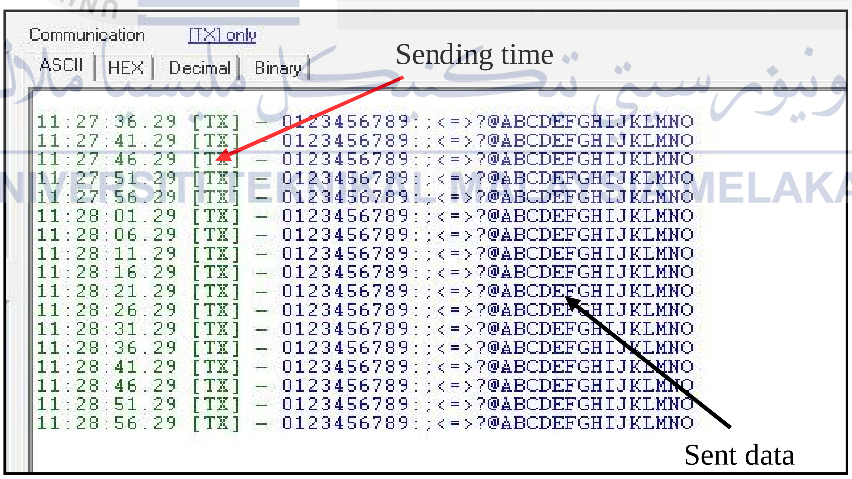


Figure 3.21: Transmitter data in Docklight monitor

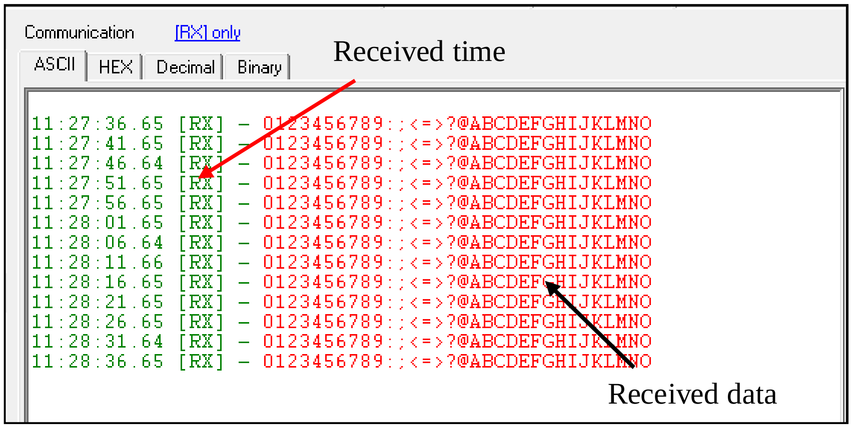


Figure 3.22: Receive data in Docklight monitor

The first step to conducting this experiment is set the clock for both sending and receiving PC at same time includes the hour, minutes and seconds. Then, place the Xbee receiver at fixed position above the ground and mark it as 0 meter of elevation. The first test is done by place the transmitter Xbee at 0 meter of distance and 1.5 meter below the receiver Xbee. The communication between the Xbee is start by click the running button in Docklight software. All the results are taken and calculate the time delay between transferring and receiving data. The experiment step is repeated for different distance that is 4m, 6m and 8m. The experiment is stop at 8 meters because it resulting a huge time delay at that position. Then, the experiment is repeated again for different elevation. To obtain the result more reliable results, the time must set perfectly for both PC.

3.4.2 Sensor performance

The objective of this experiment is to measure the best sensor output by using different resistors and the second objective is measure the repeatability and consistency of the sensor output.

The experiment parameter is the different resistors. For this experiment, 4 different resistors are selected that is 1kΩ, 10kΩ, 22kΩ, 32kΩ. The experiment is done by connecting the resistor with the Arduino microcontroller as shown in Figure 3.16. The resistor is connected in series to the ground and parallel to the input pin. However, the experiment is done by mounting the flex sensor on the glove. This because, the angle is measured based from the finger flexion.

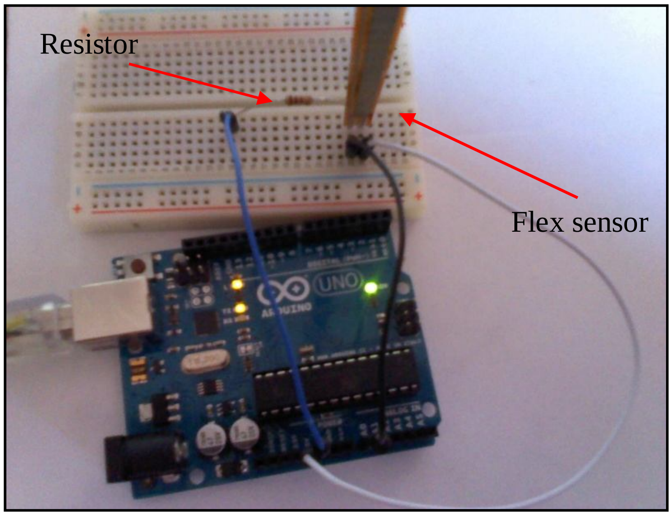


Figure 3.23: Resistor connection

From Figure 3.23, Arduino is used to read the sensor output though analog port and convert the sensor output to the voltage by using equation below and print the result in Arduino serial monitor as shown in Figure 3.25.

$$voltage = \frac{sensorValue \ x \ 5}{1023} \tag{3.1}$$

Where:

SensorValue = The analog input from the sensor (in bytes)

5 = ADC consist of 0 – 5 volts range

1023 = Decimal value range 0 – 1023 (2^{10}-1)

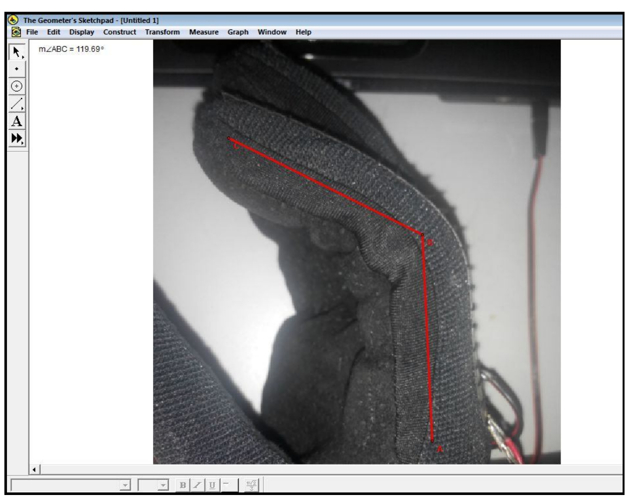


Figure 3.24: Angle of human finger at 180° - 119.69° ≈ 60°

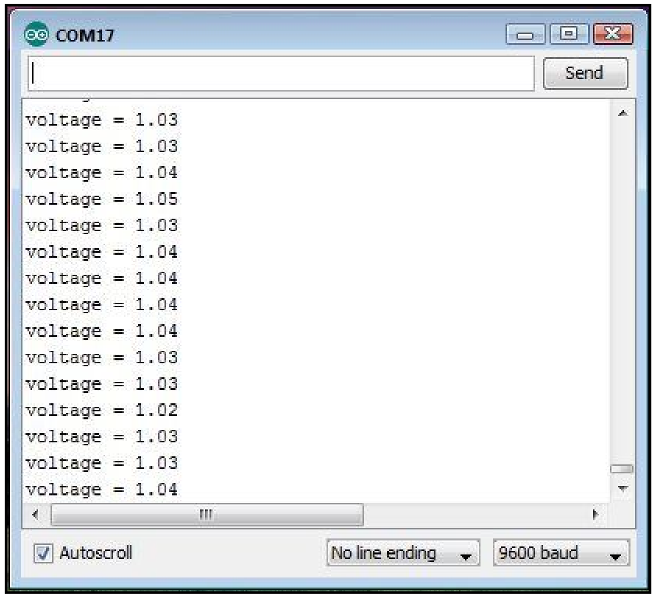


Figure 3.25: Voltage output from the Arduino serial monitor

After the required apparatus is properly setup, the first experiment is conducted by taking the output voltage when the sensor is at free position that is at 0° of finger flexion. To obtain more reliable results, the test is done for 10 times and all the result is recorded in the table. Then for the next test, the sensor is bent to 20°. The experiment is proceed with another angle of flexion that is 40°, 60°, 80° and 100°. After that, the experiment is continued with another resistor and all the steps are repeats again. The experiment procedure can be describe is Figure 3.26.

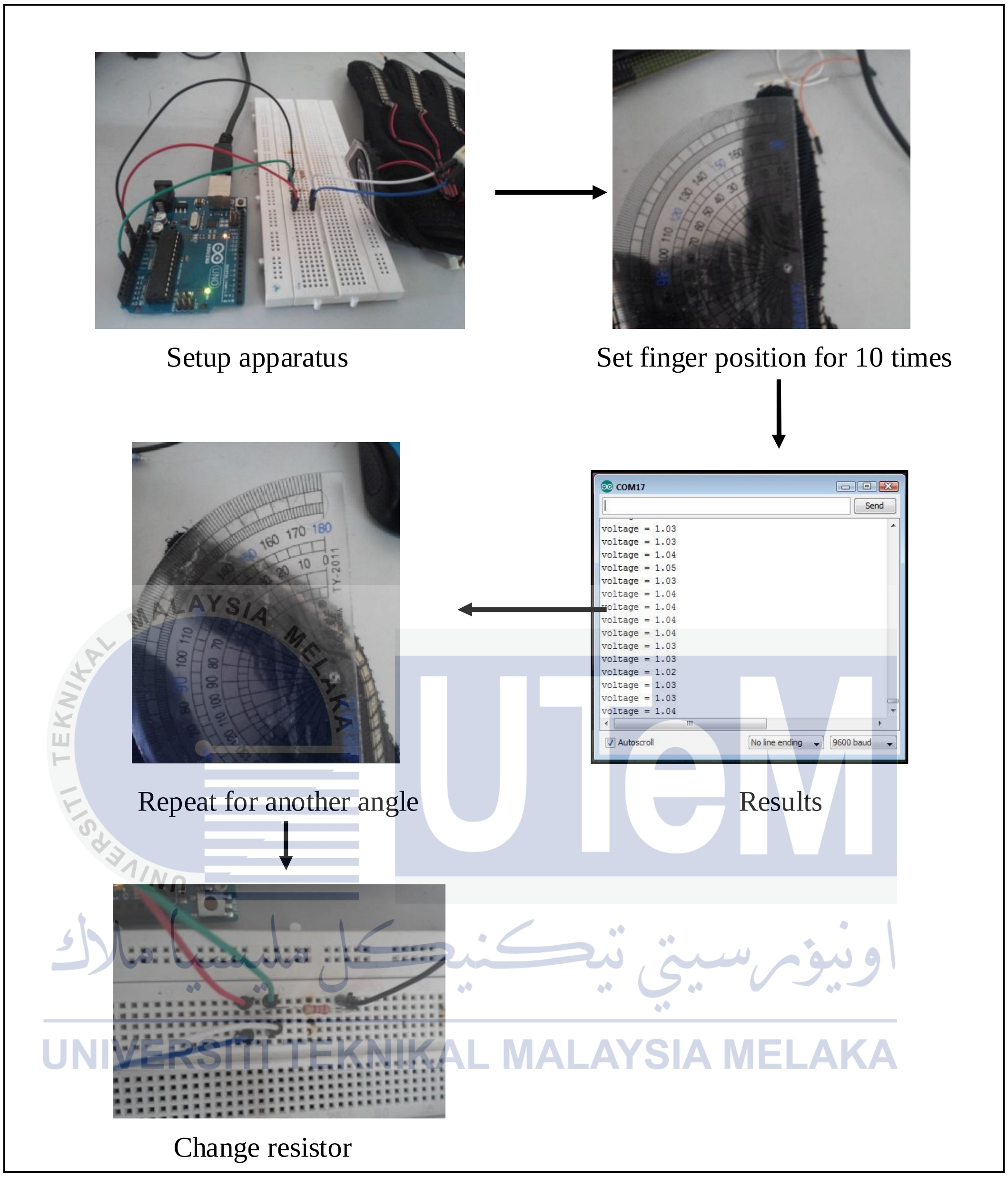


Figure 3.26: Experiment procedure

3.4.3 Spring performance

The objective of this experiment is to measure the maximum angle of the finger position when it's bent and unbent at differences hand position that is vertical, back flip, front flip. Another objective is to test efficiency, repeatability and consistency of spring.

The purpose of the spring is to return back the robot finger to its initial position, at 0° after there are no signals from the sensor. It also to make sure the robot finger is in right position as follow the position of the servo horn. The spring is mounted on the robot finger as in Figure 3.27.

Figure 3.27: Mounted spring on the robot finger

To perform this experiment, MMA7260 accelerometer is from Freescale Semiconductor is used to measure the angle of the finger position. The accelerometer used is the 3-axis accelerometer. However, the angle measurement is only used x-axis orientation of the accelerometer. This because the robot finger is flex in one orientation only. The accelerometer is mounted on the robot finger as shown in Figure 3.28. The accelerometer is only mounted for experiment purpose and not permanently.

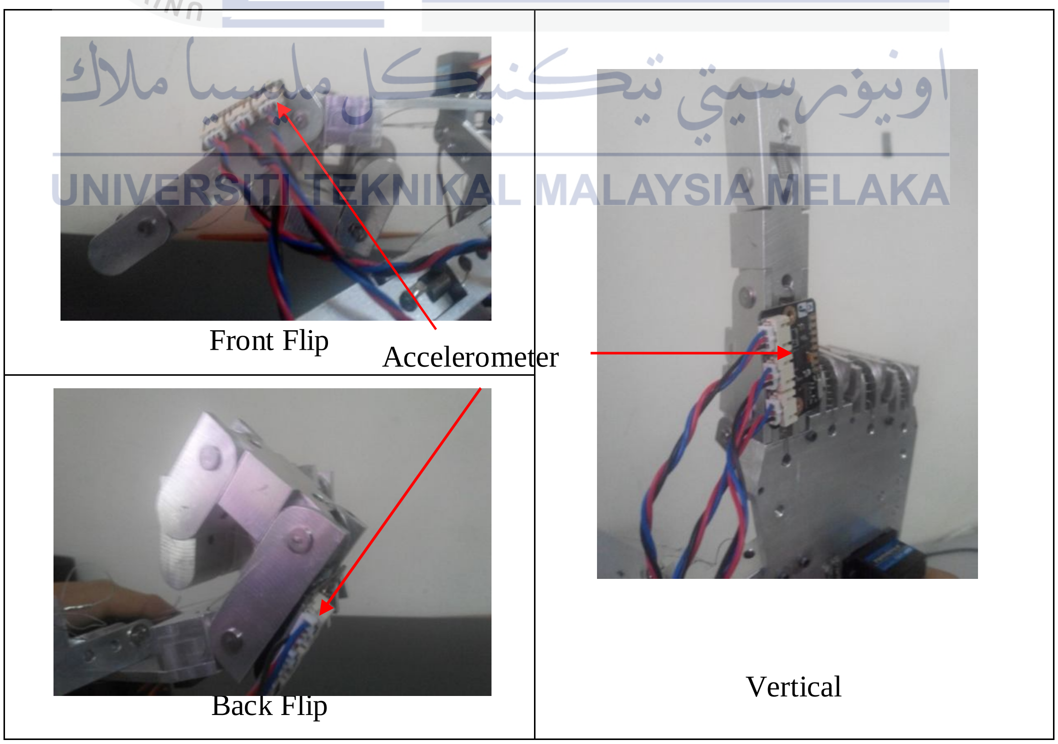


Figure 3.28: Position of the hand

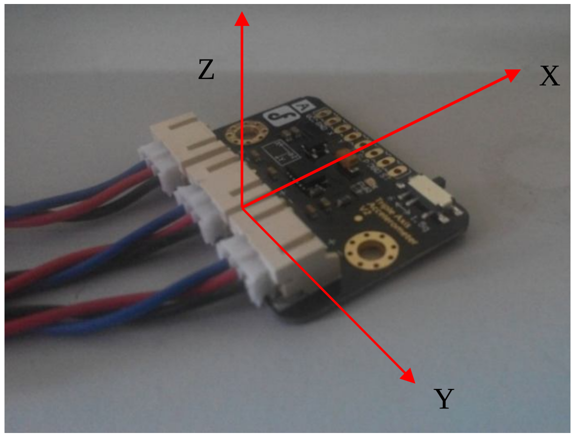


Figure 3.29: MMA7260 axis orientation

```
void loop()
{
  analog_x=analogRead(0);
  vol_x=analog_x*5.0/1024;//convert analog_x-->voltage value(v)
 //range x: 0.85 - 2.61    1.76
 //      y: 0.96 - 2.53    1.74
 //      z: 0.72 - 2.23    1.48
  add_x=vol_x-1.76;//calculate the added x axis voltage value
  g_x=add_x/0.78;//calculate the gram value

  if(g_x<=1&&g_x>=-1) //to prevent the overflow of asin(x).( If x>1 or x<-1, asin(x)=0)
  {
  degree_x=asin(g_x)*180.0/PI;//calculate the degree value
  }
  //fix the overflow condition
  if(g_x>1)
  degree_x=90;
  if(g_x<-1)
  degree_x=-90;

  Serial.print("x:");
  Serial.println(degree_x);
  delay(200);

}
```

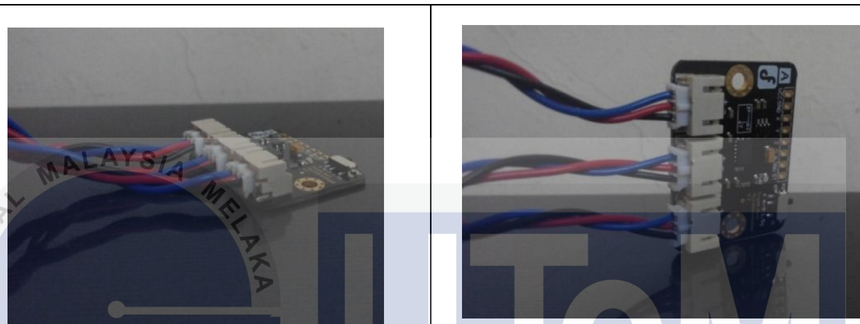

Figure 3.30: Arduino command for accelerometer

Figure 3.30 shows the command that used in the Arduino to read the analog input from the accelerometer and convert it to degree (θ°). The first step is read the input from the analog pin and then the input (in Bytes) is converted to voltage using equation (3.2). Next is calculating the voltage with x-axis voltage value. The value is the range between minimum and maximum input and after that calculates the g-value. The results can be converted into degree by using the equation (3.3). The full coding is attached in the

Appendix C. The accelerometer output range is -90° to 90° when in vertical position and the 0° output is at horizontal position. The position of the accelerometer is as shown in Figure 3.31.

$$voltage = \frac{Bytes\ x\ 5}{1024} \tag{3.2}$$

$$\theta = \frac{sin^{-1}(g - value)x\ 180}{\pi} \tag{3.3}$$



Horizontal position (0°) | Vertical position (-90° to 90°)

Figure 3.31: Accelerometer output range

Next step in this experiment is placed the robot hand in specific position as shown in Figure 3.32 as example the first test is front-flip. For each position required 2 type of test that is maximum position when performing fully flexion and maximum position when the finger is released. To obtain reliable results, the test is done for 5 times and the average value is calculated.

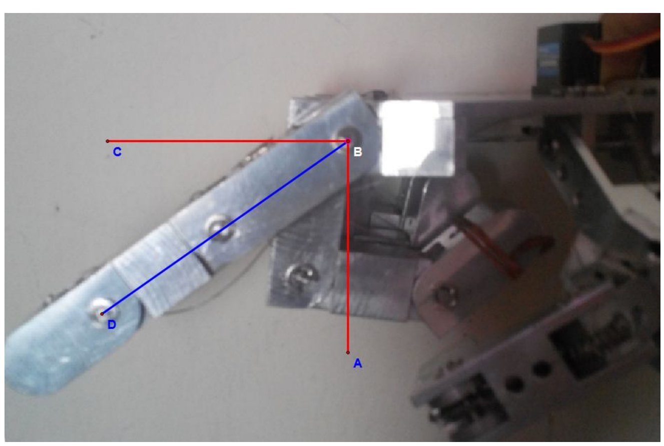


Figure 3.32: Front flip position

Figure 3.32 shows the front flip position when conducting the experiment. The angle that measured from this experiment is the angle between the reference lines, ABC with the finger position, BD where the reference line, ABC. The accelerometer measure the angle reading at position DCB. The results from the accelerometer are recorded in the table. The error of the finger can be calculated by using equation.

$$\%error = \frac{\Delta\theta}{\theta_{ref}} x100\% \qquad (3.4)$$

Where:

- $\Delta\theta$ = Angle different between measured point, D with reference point
- θref = 90°

Table 3.8: Reference point for close and open movement

Hand Orientation	Initial position of finger	
	Close	Open
Vertical	C	A
Front Flip	C	A
Back Flip	C	A

The efficiency of the finger movement is calculated using equation:

$$\%efficiency = \frac{Total\ angle, \theta_T}{Reference\ angle, \theta_{ref}} x100\% \qquad (3.5)$$

Where:

- Total angle, θ_T = Angle between initial to final position
- θref = 90°

To obtaining the best result, the hand palm must place on fixed and flat surface area without any slanted surface. Make sure also the hand palm is in 0° angle when performing back-flip and front-flip test and 90° angle when perform the vertical test.

3.4.4 Measurement of finger position

The objective of this experiment is to measure the angle of the robot finger that corresponds with human fingers. The second objective is to measure the error of robot finger flexion.

This experiment requires master data glove, robot hand prototype and accelerometer to measure the angle. The experiment is done to all fingers but it is focused on two desired angle only which are 45° and 90°. For this experiment, the angle is measured between proximal and middle phalanx and the accelerometer is mounted on the distal phalanx as shown in Figure 3.33.

Figure 3.33: Position of the accelerometer on the finger

From Figure 3.33, the accelerometer is mounted with different orientation if compared to the experiment 2 because the finger has small surface area. The angle reading for this experiment is obtained from y-axis of the accelerometer. So, the accelerometer coding from experiment 2 needs to modify so that it can read output from the y-axis orientation instead of x-axis. The changes are on the range part and the analog pin. The full coding is attached on Appendix D.

```
void loop()
{
  analog_y=analogRead(1);

  vol_y=analog_y*5.0/1024;//convert analog_y-->voltage value(v)
  add_y=vol_y-1.74; //calculate the added x axis voltage value
  g_y=add_y/0.8;//calculate the gram value

  if(g_y<=1&&g_y>=-1) //We use this condition to prevent the overflow of asin(y).( If y>1 or y<-1, asin(y)=0)
  {
  degree_y=asin(g_y)*180.0/PI;
  }
  //fix the overflow condition
    if(g_y>1)
  degree_y=90;
    if(g_y<-1)
  degree_y=-90;
}
```

Figure 3.34: Arduino coding to read analog input from y-axis

The experiment is started with bending the human finger to desired angle that is 45°. The accelerometer reading is taken after the robot hand not moving and in stable position. The result can be obtained from the Arduino serial monitor. The experiment is repeated 5 times for each desired angle to obtain the average value. All the data is then tabulated on the table. Error of the finger is calculated by using the equation.

$$\%\ error = \frac{\theta h - average\ \theta r}{\theta h} x100 \qquad (3.6)$$

Where: θr = mean value for angle of robotic finger.

θh = angle of human finger

As the precaution in obtaining the best result, the accelerometer cannot be slanted when mounting it on the finger. As example, the value is measured from y-axis, so the initial position of the accelerometer is on 0° of y-axis and 0° x-axis. If the position on the x-axis is slanting, it will cause an error on y-axis output value.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Robot hand design

The prototype of the robotic hand has completely design by using SolidWorks software. The robotic prototype is designed with 5 Degree of Freedom (DOF) where 1 DOF for each finger. The design is shown in Figure 4.1 while the isometric view of the design is attached in the Appendix F. For the electrical system, the system has been designed by using Fritzing software. Schematic and circuit diagram is completely designed for both transmitter and receiver system. The transmitter system is shown in Figure 4.2 and Figure 4.3 while the receiver system is shown in Figure 4.4 and Figure 4.5.

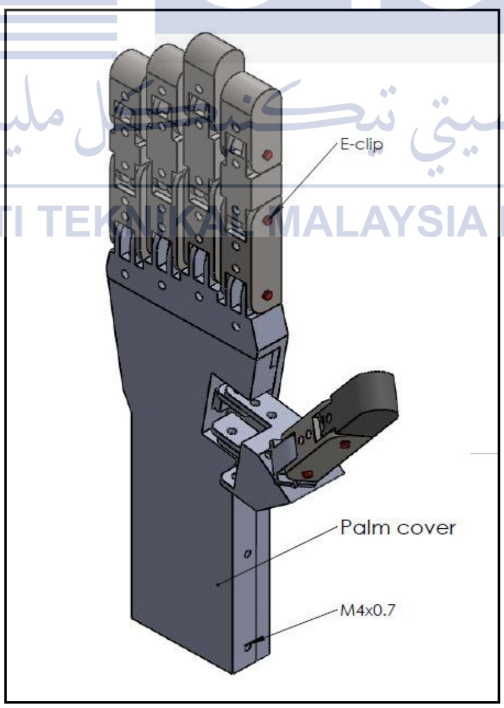


Figure 4.1: Complete design of robotic hand

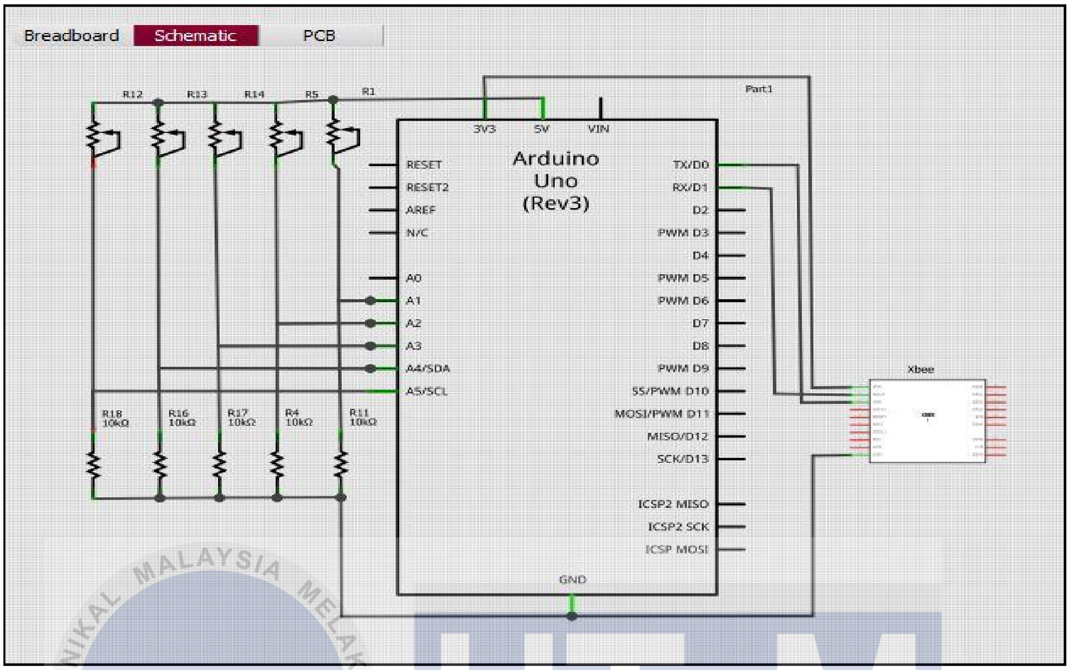


Figure 4.2: Schematic design for transmitter system

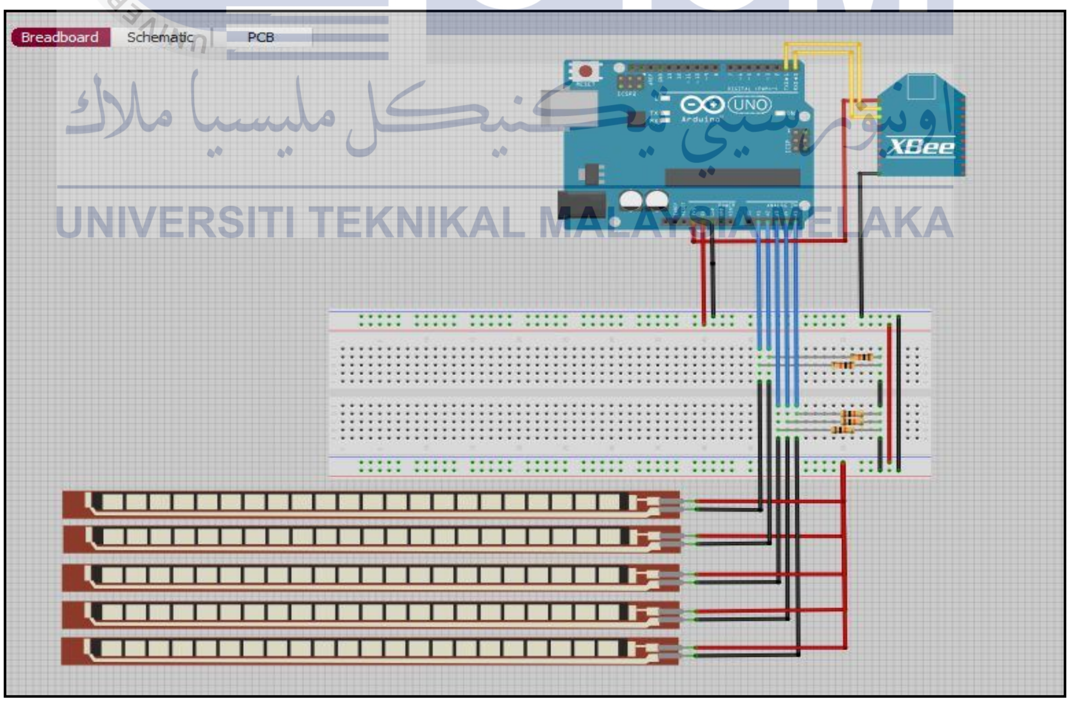


Figure 4.3: Circuit design for transmitter system by using Fritzing software

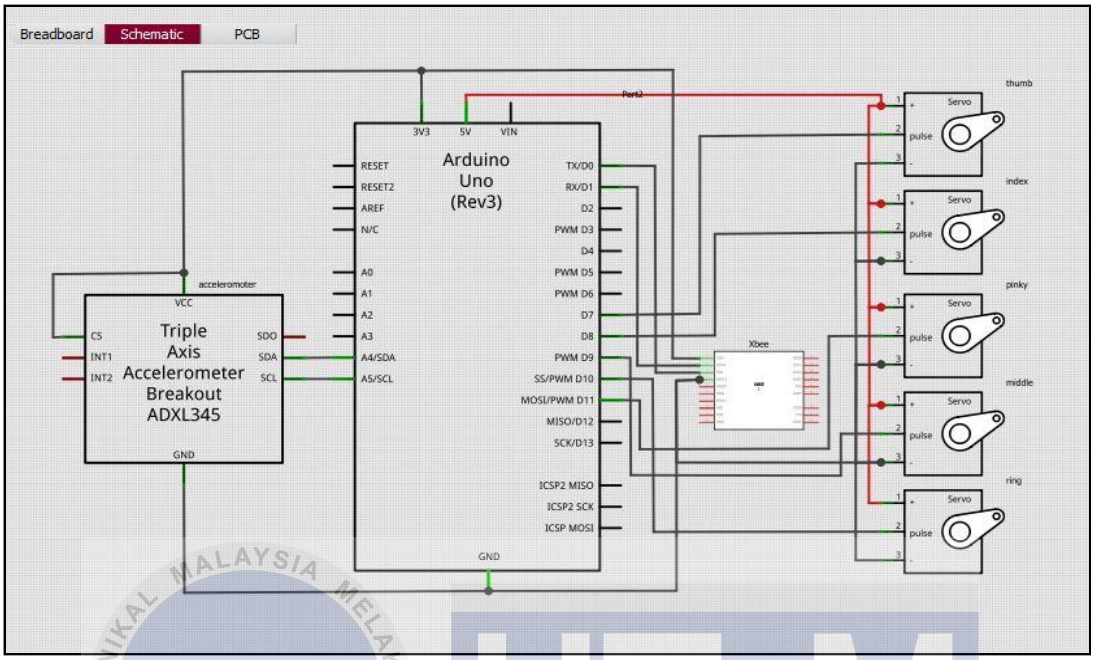


Figure 4.4: Schematic design for receiver system

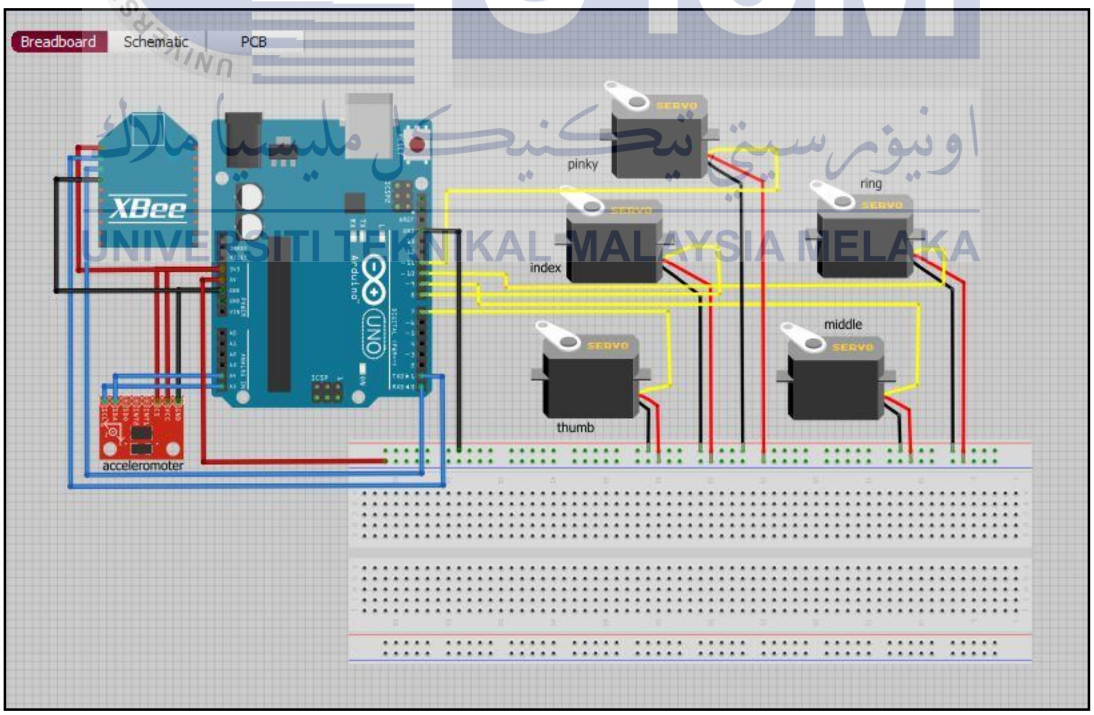


Figure 4.5: Circuit design for receiver system by using Fritzing software

4.2 Robot Hand Prototype

The robot hand has successfully developed and fully functioning. The materials used for the robot hand development is the aluminum 6061. Figure below is the full pictures of the robot hand after the fabrication and fully assemble together with the actuator system. The actuator system is use the servo motor. The servo motor will rotates and pull the robot finger by using a wire. The wire that use to pulling the robot finger is nylon coated wire. This type of wire is very flexible, soft and strong. The selected wire for this project has thickness with 0.25mm and can support load up to 10lb (4.536kg).

The robot hand is operating by mimic the human finger flexion where the data signal that read from the sensor is sent through the serial communication via Xbee module.

Figure 4.6: Robot hand prototype

Figure 4.7: Circuit system

4.3 Sensor output

The sensor output is obtained in order to complete mapping command in the Arduino. The mapping command required 2 values that are “fromLow” and “fromHigh”. The output value from the flex sensor is obtained after done the Experiment 2 that is experiment on sensor performance. This because the required value from the sensor is the value after added the resistor. To obtain the exact result, all the flex sensors must be mounted on the glove because the angle measured is based on the finger flexion. The value that gets from the sensor is value when the finger in 0° of flexion and another value is at 180° of flexion. All the sensors must undergo this process because each of them has different value and range as shown in Table 4.1.

Table 4.1: Sensor value

Finger	Sensor value (Bytes)		
	0°	180°	Range
Thumb	435	270	165
Index	420	170	250
Middle	400	220	180
Ring	540	430	110
Pinky	450	340	110

However, this sensor value will cause the servo motor make overshoot turning either in 0° position or 180°. The overshoot problem will make the servo gear jammed. To overcome this problem, the sensor value range is widened by adding 160 Bytes of data. The value in 0° is added with 80 Bytes and value in 180° is minus with 80 Bytes. The added value is obtain by testing on the servo and the data that not cause overshoot problem is taken. However the added value is not valid to the thumb. It is because, the thumb use 2 servos as the actuator. So, the length of wire required to pull the thumb is short compared than the other finger. Then, the sensor output range is set with large value to make sure the servo rotation is not too sensitive finger flexion. If the finger fully bent, the will not rotate up to 180° and it only rotate to the needed angle. After make some adjustment on the sensor mapping, the best value has been found where the value at 0° of finger flexion is added with 80 Bytes while value at 180° of finger flexion is minus with 190 Bytes. The final mapping value is as shown in Table 4.2.

Table 4.2: Sensor value after modification

Finger	Sensor value (Bytes)		
	0°	180°	Range
Thumb	515	80	435
Index	500	90	410
Middle	480	140	340
Ring	620	350	320
Pinky	530	260	270

4.4 Experimental results

This chapter will show the results of this project. The result is measured from the conducted experiments and tabulated in table form and graph. Based on the results, the performance of the robotic finger is analyzed and discussed.

4.4.1 Wireless communication performance

This experiment is to analyze the performance of wireless communication. The test subject is the Xbee module by obtaining the time delay for data transfer between two Xbee over a distance. For this project, the experiment is done in indoor since the robot is for indoor purpose. At the end of this experiment, the maximum position for the transmitter placement that gives best data transfer will be obtained. All the result of delay time has been put in Table 4.3.

Table 4.3: Time delay over a distance

Elavation (m)	Distance (m)	Time Delay (s)					
		1st	2nd	3rd	4th	5th	Average
-1.5	0	0.22	0.23	0.23	0.22	0.22	0.22
	2	0.37	0.36	0.36	0.37	0.37	0.37
	4	0.44	0.44	0.43	0.44	0.44	0.44
	6	0.62	0.61	0.62	0.61	0.62	0.62
	8	0.84	0.85	0.85	0.84	0.84	0.84
0	0	0.19	0.18	0.18	0.19	0.19	0.19
	2	0.28	0.28	0.27	0.27	0.28	0.28
	4	0.36	0.36	0.35	0.36	0.36	0.36
	6	0.51	0.49	0.48	0.49	0.49	0.49
	8	0.61	0.60	0.60	0.61	0.60	0.60
1.5	0	0.16	0.16	0.15	0.15	0.16	0.16
	2	0.24	0.25	0.25	0.24	0.24	0.24
	4	0.33	0.32	0.33	0.33	0.32	0.33
	6	0.45	0.45	0.44	0.44	0.45	0.45
	8	0.56	0.56	0.55	0.55	0.56	0.56

The test is done for point-to-point communication and 32 Bytes of data is set for this performance test. The data has been setup is ASCII mode.

Set data:

0123456789:;<=>?@ABCDEFGHIJKLMNO = 32 Bytes/Transfer

The data sequence is sent repeatedly for every 5 seconds. From the experiment, 32 Bytes of data is successfully receive by the receiver and it will record the time for every receive data as shown in Figure 4.8 where the green color is the time and red color is the receive data.

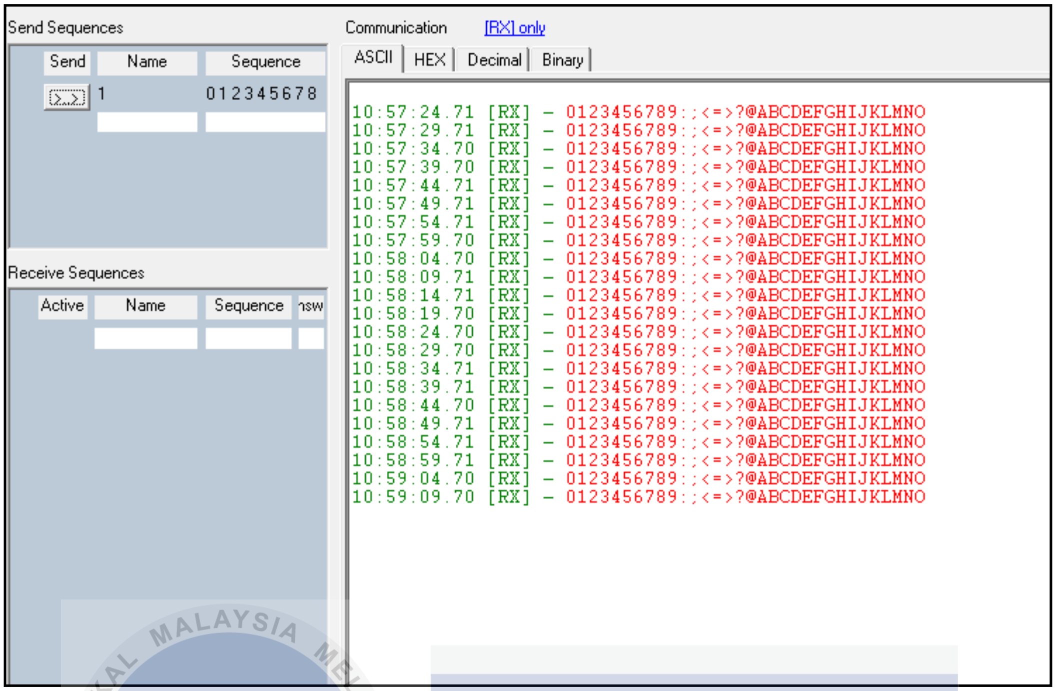


Figure 4.8: Receive 32 Bytes of data

From the result in Table 4.1, the best range in order to performing the data transfer more efficiently is between 0 to 8 meters. However, the 8 meter distance is more suitable for 1.5 of elevation. This because, the Xbee have some delay when initiate the connection between them at 0 elevation and below than that. This delay problem can be proved in Figure 4.9 for the transmitter and Figure 4.10 for the receiver.

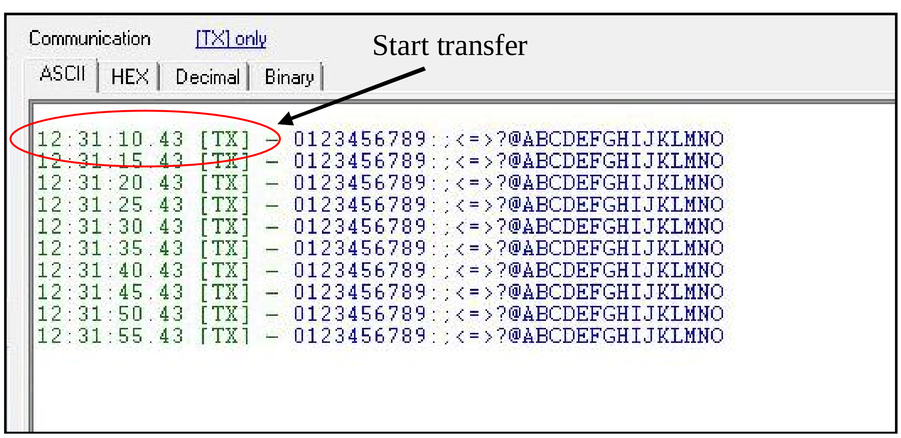


Figure 4.9: Data transfer at 8 meter, 0 elevations

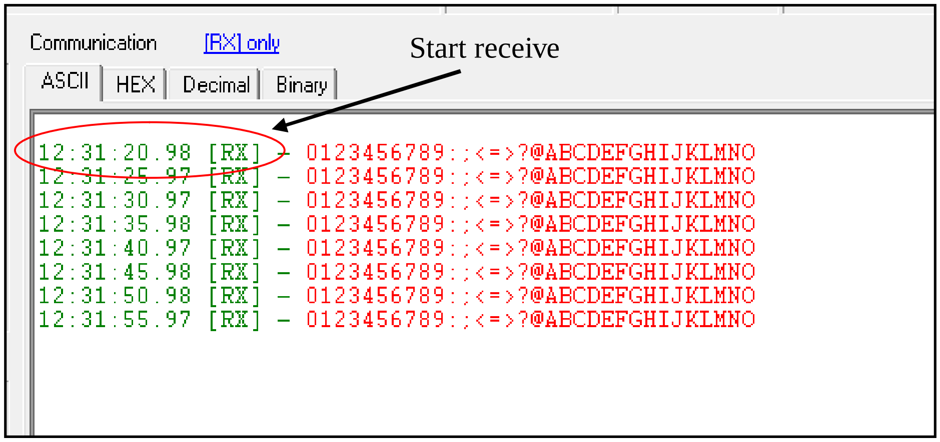


Figure 4.10: Data receive at 8 meter, 0 elevations

Figure 4.9 and Figure 4.10 shows the delay when initiating the connection when the Xbee transmitter is placed 8 meter from the receiver. At this time the elevation level is 0 meter and the time delay is 10.55 seconds before the connection is successful. After the connection has successful, the data can transfer in sequences but it still has a delay for each data transfer.

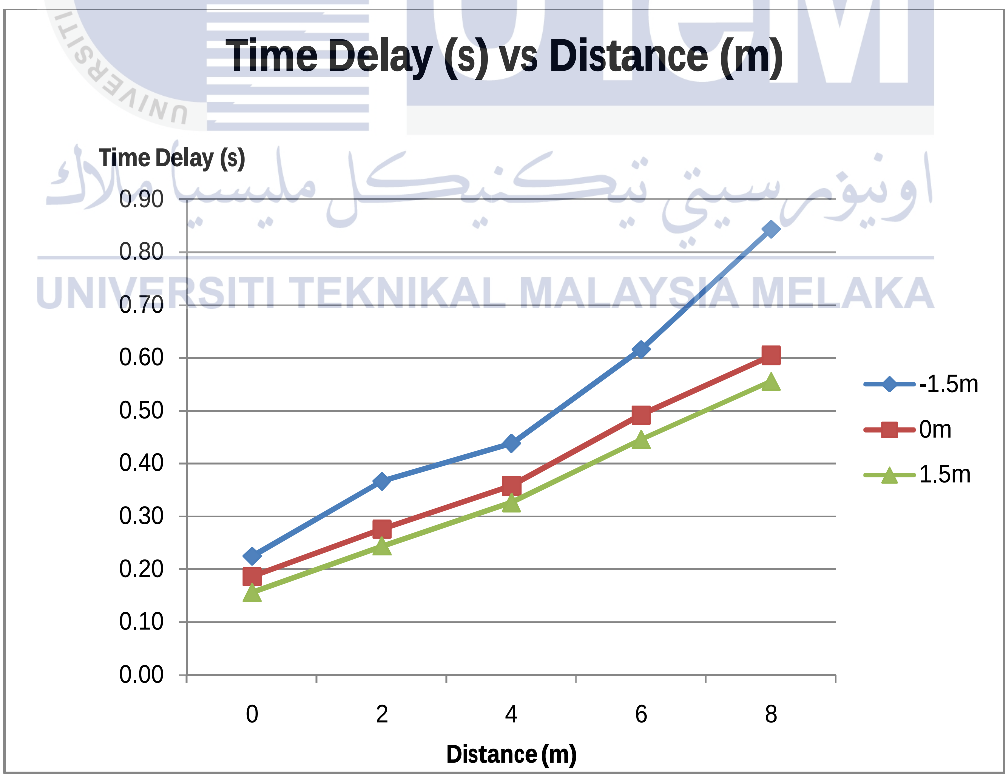


Figure 4.11: Graph Time delay versus distance

From the graph above in Figure 4.11, the best elevation level between the transmitter and the receiver are at 0 meter and 1.5 meters. Although 1.5 meter of elevation has better data transfer compared to 0 meter elevation, but the time delay different is too small that around ± 0.04 seconds. This small different not give too much effect on the data transfer. For -1.5 meters of elevation, the data transfer is too bad because too many delay. This is happen because the test is done in indoor and there are many obstacles that cause the wireless signal become weak and make the data difficult to transfer to another Xbee.

4.4.2 Sensor performance

This experiment is to test and analyze the performance of flex sensor by using different resistors. Resistor the gives best results is used with the flex sensor. The test parameter is the output voltage of the sensor for different flexion angle. All the result is tabulated in a specific table.

Table 4.4: Output voltage for 1kΩ resistor

	No of Trials	Angle (θ°)					
		0	20	40	60	80	100
voltage (v)	1	0.38	0.33	0.27	0.22	0.18	0.16
	2	0.37	0.33	0.28	0.23	0.20	0.15
	3	0.37	0.31	0.28	0.22	0.18	0.15
	4	0.39	0.32	0.28	0.22	0.18	0.15
	5	0.39	0.32	0.29	0.23	0.19	0.16
	6	0.38	0.32	0.29	0.23	0.19	0.15
	7	0.36	0.33	0.30	0.21	0.20	0.16
	8	0.37	0.34	0.29	0.21	0.18	0.15
	9	0.38	0.32	0.29	0.21	0.18	0.15
	10	0.37	0.32	0.29	0.22	0.19	0.15
	Average	0.376	0.324	0.286	0.22	0.187	0.153

Table 4.5: Output voltage for 10kΩ resistor

	No of Trials	Angle (θ°)					
		0	20	40	60	80	100
Voltage (v)	1	2.23	2.00	1.78	1.52	1.37	1.22
	2	2.23	1.99	1.76	1.55	1.36	1.23
	3	2.24	1.98	1.77	1.53	1.35	1.23
	4	2.22	1.97	1.78	1.54	1.36	1.22
	5	2.23	1.99	1.76	1.52	1.38	1.22
	6	2.23	1.98	1.79	1.52	1.35	1.23
	7	2.22	1.99	1.76	1.54	1.37	1.21
	8	2.21	2.00	1.76	1.54	1.37	1.23
	9	2.22	1.99	1.79	1.53	1.36	1.22
	10	2.23	1.98	1.77	1.55	1.37	1.23
	Average	2.226	1.987	1.772	1.534	1.364	1.224

Table 4.6: Output voltage for 22kΩ resistor

	No of Trials	Angle (θ°)					
		0	20	40	60	80	100
Voltage (v)	1	3.13	3.07	2.76	2.59	2.35	2.03
	2	3.12	2.97	2.76	2.58	2.33	2.05
	3	3.11	2.99	2.76	2.57	2.37	2.04
	4	3.11	2.97	2.71	2.59	2.36	2.05
	5	3.12	2.98	2.73	2.57	2.34	2.04
	6	3.11	2.87	2.75	2.58	2.38	2.04
	7	3.13	3.05	2.76	2.56	2.36	2.05
	8	3.12	3.03	2.74	2.60	2.34	2.03
	9	3.12	2.99	2.74	2.60	2.35	2.06
	10	3.11	3.01	2.72	2.58	2.38	2.02
	Average	3.118	2.993	2.743	2.582	2.356	2.041

Table 4.7: Output voltage for 32kΩ resistor

	No of Trials	Angle (θ°)					
		0	20	40	60	80	100
Voltage (v)	1	3.60	3.36	3.20	3.02	2.94	2.64
	2	3.62	3.39	3.18	3.09	2.95	2.61
	3	3.62	3.4	3.25	3.05	2.93	2.6
	4	3.61	3.38	3.17	3.02	2.96	2.63
	5	3.59	3.37	3.19	3.01	2.92	2.61
	6	3.59	3.39	3.20	3.04	2.90	2.62
	7	3.60	3.36	3.21	3.06	2.93	2.61
	8	3.58	3.38	3.20	3.03	2.93	2.6
	9	3.61	3.37	3.21	3.00	2.91	2.64
	10	3.61	3.37	2.23	3.00	2.89	2.61
	Average	3.603	3.377	3.104	3.032	2.926	2.617

This experiment covers sensor repeatability test and consistency test. In order to obtain accurate results, 10 trials are done for each test. The test is done to measure the best voltage output from the flex sensor. The sensor test is performed together with Arduino microcontroller where the output voltage can easily obtain by monitor the result using the Arduino serial monitor. Arduino will read the analog input data from the sensor in Bytes but in can be converted to voltage by using command as shown in Figure 4.12.


```
void loop()
{
  int sensorValue = analogRead(A0); // read the input
  float voltage = sensorValue * (5.0 / 1023.0);  // Convert the analog reading to voltage
  Serial.println(voltage);
}
```

Figure 4.12: Command to convert analog input to voltage

From the command, the voltage reading can be obtained from the serial monitor of the Arduino software. All the data obtain from the experiment is applied into a graph as shown in Figure 4.13.

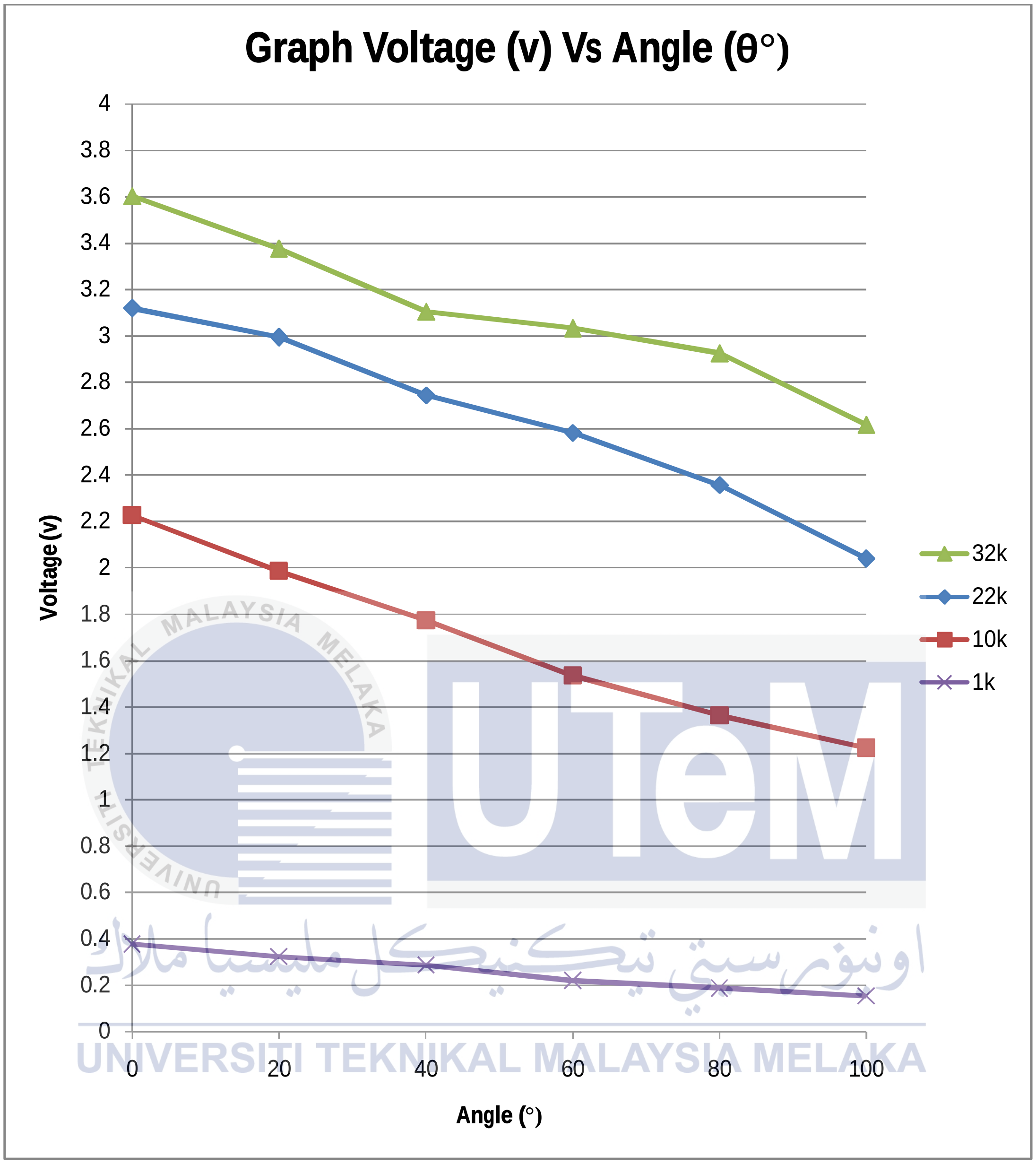


Figure 4.13: Graph Voltage versus Angle for differences resistor

The experiment is done by measuring the angle of finger flexion. From the graph, the sensor output is declined from 0^o to 100^o. This because of the flex sensor is bend backward the when performing the experiment. From the results, 10kΩ resistor shows the most suitable resistor that can be used with the flex sensor. This because it has consistent changes for every 20° of flexion compared to other resistor where there are small different in gradient for each segment as shown in Table 4.8.

Table 4.8: Gradient, M value for every 20° of flexion

Angle (θ°)	Gradient, $M = \left\|\frac{Y_2-Y_1}{X_2-X_1}\right\|$			
	1kΩ	10kΩ	22kΩ	32kΩ
0-20	0.0026	0.0119	0.0063	0.0113
20-40	0.0019	0.0107	0.0125	0.0137
40-60	0.0033	0.0119	0.0081	0.0036
60-80	0.0017	0.0085	0.0113	0.0053
80-100	0.0017	0.0007	0.0158	0.0155

Although the resistor 1kΩ also almost gives constants change like 10kΩ resistor, but it has very small output range that only |0.153v-0.376v|=0.223v while 10kΩ resistor has range |1.224-2.226|=1.002v. High range of voltage output will cause the sensor more sensitive if compared to low range output and it can read the every flexion changes more accurate and stable. By comparing the 10kΩ with 22kΩ resistor, both resistors have wide range that exceeds more than 1. However, 22kΩ resistor has inconsistent changes for every 20° of flexion as shown in the graph. From the 4 resistors, 10kΩ has good results in repeatability test compared to other where the results can be seen in table 4.2. To perform this repeatability test, 10 trials is done for every angle of finger flexion. From table 4.2, it shows the sensor have stable output for every angle where the output different for each test is around ±0.01v. This means, 10kΩ resistor is good resistor to use with the sensor in order to obtain high repeatability performance.

4.4.3 Spring performance

In this experiment, two tests are carried out to test the spring performance for differences hand position that is vertical, front flip and back flip. The spring is tested by measuring the maximum angle of finger position that cover both close and open finger flexion.

Table 4.9: Measured angle for close and open movement

		Hand Position					
		Vertical		Front Flip		Back Flip	
No of Trials	Finger Flexion	Close	Open	Close	Open	Close	Open
1	Total distance (θT)	88.52	90.00	88.74	56.08	38.54	90.00
2		88.69	90.00	87.91	59.56	68.36	90.00
3		88.73	90.00	88.28	57.62	30.65	90.00
4		87.96	90.00	88.76	55.85	36.28	90.00
5		88.38	90.00	87.85	56.28	34.71	90.00
Average		88.46	90.00	88.31	57.08	41.71	90.00
Δθ		1.54	0.00	1.69	32.92	48.29	0.00
% Error		1.72	0.00	1.88	36.58	54.37	0.00
%Efficiency		98.28	100.00	98.12	63.42	53.66	100.00
*Δθ = Reference θ (90°) - Measured θ *Total distance, θT = Angle difference between initial and final position							

From the Table 4.9, it shows that the robot finger can fully closed in vertical and front hand of hand position while it can perform the fully open in vertical and back flip of hand position. The result also shows that the finger has consistent performance in close movement for vertical and front flip position. But, for open movement, the robot finger only has consistent performance in vertical and back flip position. Bar chart is plotted based on the average value for both close and open movement as shown in Figure 4.14.

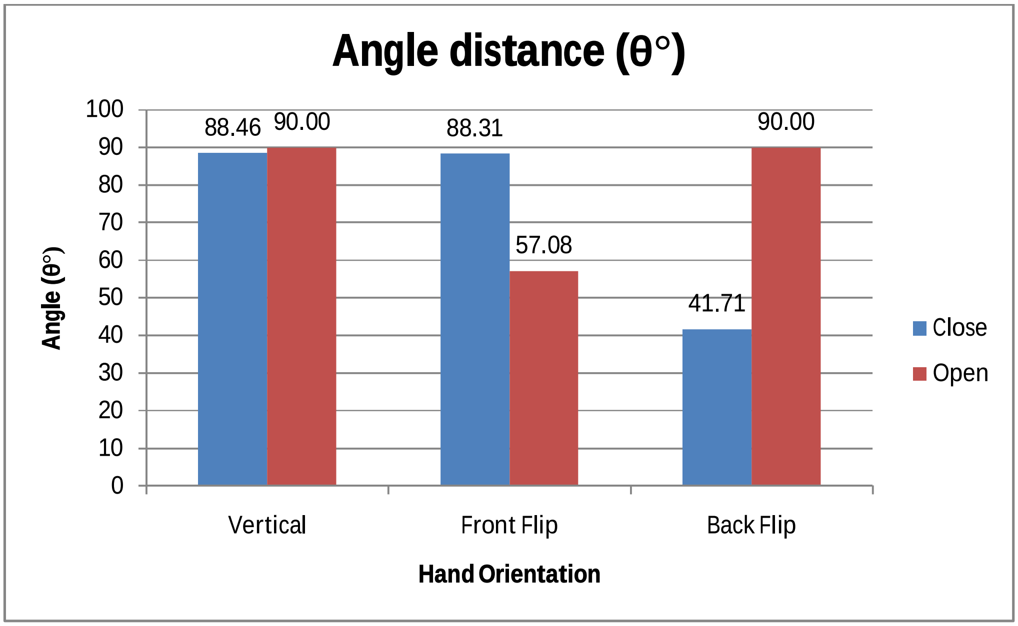


Figure 4.14: Bar chart of the hand orientation

From the bar chart, it concludes that the best position for the spring work perfectly is the vertical position. This because the chart shows that the robot finger can bend up to its maximum positions either in close or open movement. However, the finger only can bend up to 88.456° when it performs close movement due to the error during the fabrication process that cause the finger can't reach 90° of flexion. This fabrication error only happens for the close flexion. The spring has bad performance when the hands in front flip orientation. These because the spring doesn't have enough strength to full back the finger to its open position where it only can move to 57.08° from its initial position. This problem is because of the finger is too heavy for the spring and another factor that cause this problem is the gravitational factor. Spring with high constant value, K is required to counter back this problem. Another performance problem from experiment is when the robot hand at back-flip position. The problem is the finger cannot perform the close movement. It's also due to the gravitational problem since the hand is placed in horizontal position. Moreover, the wrong calculation on the selection of the actuator is the major causes to this problem. This because the servo doesn't have enough torque to pull up the finger although at time the selected servo can support the torque up to 1.8kg/cm (4.8V) - 2.2kg/cm (6.0V). The best way to counter this problem is change the current servo to more high torque servo.

4.4.4 Measurement of finger position

This experiment is to measure the angle position of robotic finger. This experiment is focused on two reference angle only which is 45° and 90°. The results for 45° of reference angle are tabulated into Table 4.10 while the results for 90° of reference angle are tabulated into Table 4.11.

Table 4.10: Comparison of finger flexion for 45° of reference angle

Reference Angle (45°)							
Finger		1st	2nd	3rd	4th	5th	Average
Thumb	Human	43.22	45.03	45.65	48.72	46.58	45.84
	Robot	50.66	52.84	53.41	55.76	53.95	53.32
Index	Human	46.63	47.66	44.78	46.97	47.93	46.79
	Robot	36.65	37.49	36.69	37.13	38.72	37.34
Middle	Human	46.24	44.86	44.52	47.17	48.38	46.23
	Robot	39.94	37.22	39.49	40.27	41.57	39.70
Ring	Human	46.03	47.24	44.93	45.76	45.16	45.82
	Robot	39.82	41.14	38.43	38.56	38.90	39.37
Pinky	Human	42.11	42.37	46.65	54.77	38.63	44.91
	Robot	55.58	56.22	62.33	77.23	50.87	60.45

Table 4.11: Comparison of finger flexion for 90° of reference angle

Reference Angle (90°)							
Finger		1st	2nd	3rd	4th	5th	Average
Thumb	Human	71.55	68.93	67.14	66.92	69.57	68.82
	Robot	83.34	80.89	78.93	76.54	81.54	80.25
Index	Human	90.45	91.73	90.38	90.81	90.72	90.82
	Robot	87.49	88.84	88.19	87.8	87.54	87.97
Middle	Human	90.39	89.64	91.14	90.53	90.72	90.48
	Robot	87.35	87.83	88.88	87.04	88.92	88.00
Ring	Human	90.03	90.28	90.51	91.02	90.85	90.54
	Robot	86.86	87.39	87.68	88.97	88.06	87.79
Pinky	Human	90.63	91.25	89.05	90.61	91.79	90.67
	Robot	86.95	87.02	86.73	86.69	87.06	86.89

The experiment covers the accuracy test that compared the accuracy of finger flexion between robot finger and human finger. The result is applied to the bar chart as shown in Figure 4.15 for 45° of reference angle while Figure 4.16 for 90° of reference angle.

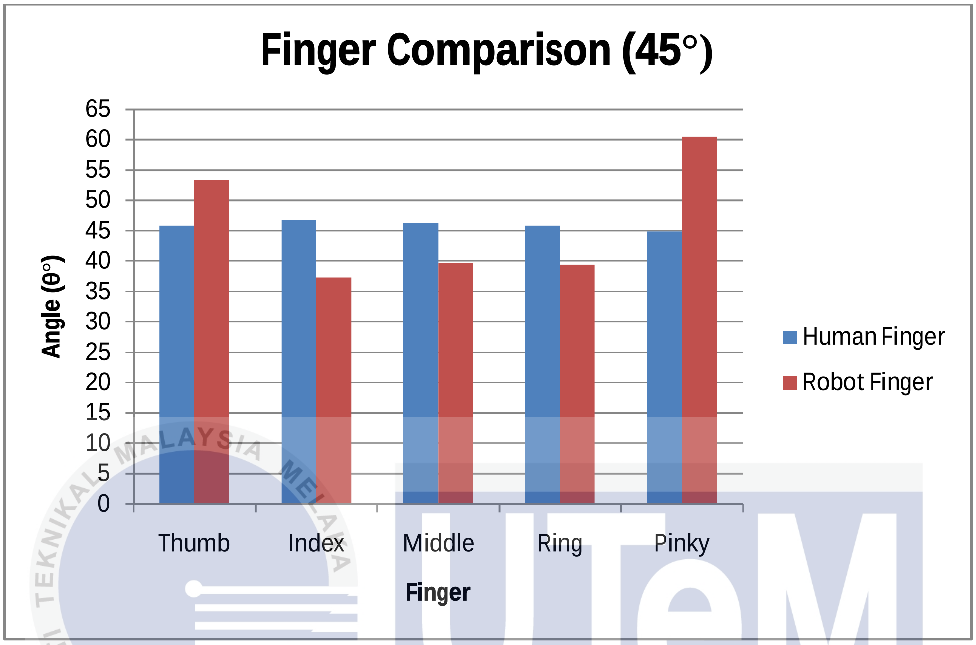


Figure 4.15: Comparison of finger flexion for 45° of reference angle

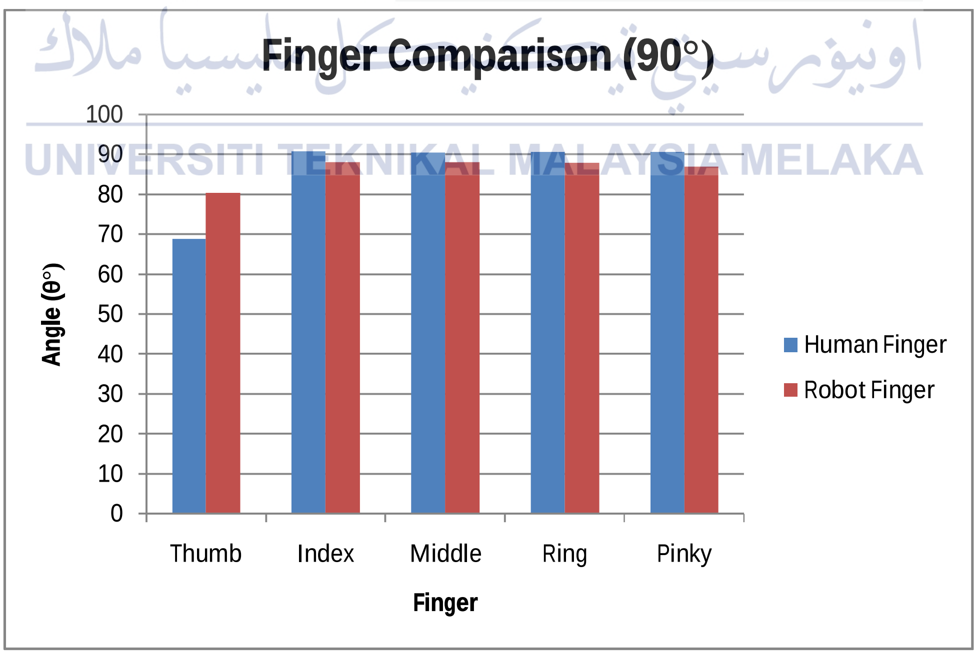


Figure 4.16: Comparison of finger flexion for 90° of reference angle

The result shows that there is slightly different value of angle reading between robot and human finger flexion 90° of reference angle while for 45° of reference angle shows a large different between robot and human finger. The error between the robot finger flexion and human finger flexion can be calculated by using equation (4.1).

$$\%error = \left|\frac{\theta_h - average\theta_r}{\theta_h}\right| x100 \qquad (4.1)$$

Where: θr = angle of robot finger

θh = angle of human fingers

Table 4.12: Percentage of error between robot and human finger flexion

Reference angle (θ°)	%error				
	Thumb	Index	Middle	Ring	Pinky
45°	16.33	20.21	14.14	14.08	34.61
90°	16.60	3.13	2.69	3.03	5.16

For 45° of reference angle, pinky finger shows the highest error compared to other finger. This because the pinky has small design compared to other finger. Because of that, pinky finger only require 56.7mm of cable to fully actuate. However, the servo horn used has 29.27mm of length as shown in Figure 4.17.

Figure 4.17: Length of the servo horn

To perform the actuation, the servo will produce 29.27X2=58.54mm from 0° to 180° of servo rotation which means it has exceeded the requirement needed by the pinky finger. This problem cause the pinky finger has bent a little bit more than required flexion. Another finger almost have the same angle value if compare between the robot and human finger because the error is less than 20%. It is difficult to have accurate angle of flexion like human finger flexion especially in 45° degree of reference angle because each finger segment is actuate together that only have one DOF.

For 90° of reference angle, the robot finger produce nice result since the error of finger flexion is small that below than 5%. Although the error is happen, the robot finger is actually already bent at its maximum bending. The error happened when the robot hand undergoes fabrication process where it can't fabricate 100% same as the design by Solidworks. The results also show the problem at the thumb where it has large error compared to other finger that is 16.6%. This because human thumb can't bend up to 90° but the robot thumb is required to bend up to 90°. That's why the thumb has large error between the robot and human. Robot thumb can be actuate up to 90° by controlling the servo up to its limit that at 180° of rotation. Although human thumb cannot bend to 90°, servo still can be controlled by mapping the sensor value at the command as shown in Figure 4.18.

```
//mapping sensor value
int value1 = map(pin1, 200, 620, 0, 179); // (6)
```

Figure 4.18: Mapping sensor value

Based on the mapping command, pin1 is the assigned pin to read analog input. 200 is the minimum output from the sensor while 620 is the maximum output. The sensor output is in Byes. 0 and 179 is to set the initial and final position of the servo. If the value is set in reversed, the servo motor also will rotate is reverse direction.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

This section consists of the conclusion and recommendation which is the last part of this project. The conclusion is discussed about the progress of this project that based on the objective that stated in Chapter 1. Recommendation part is discussed for continuation and further development of this robotic hand project.

5.1 Conclusion

The first objective of this project is partially achieved when the prototype of the robotic hand was completely designed by using SolidWorks software. The electrical designs that cover the transmitter and the receiver system also have been designed by using Fritzing software. The robot hand has successfully fabricated by using aluminum for overall part. The electrical circuit has been made by soldering some component on the breadboard where the board can easily mounted on the Arduino without any extra soldering.

The electrical system that include the master data glove and the robot system also have successfully developed. The communication between the data glove with robot system can synchronize successfully though wireless communication via Xbee module. The robot actuator can be control perfectly over a distance wirelessly where the signal from the sensor is received by the robot through the Xbee module and its means the objective two has achieved. The major component on the data glove are Arduino microcontroller, flex sensor and Xbee module while for the robot actuating system are Arduino microcontroller, servo motor and Xbee module.

The last objective is about the robotic hand performance test that includes the wireless communication performance, spring performance, sensor performance and measurement of finger position. The objective is completely achieved because all the experiment that covers the testing and analyzing the robot hand performance is completely done. The experiment outline is as state in Chapter 3 that include the objective, experiment setup and procedure. The result that obtained from the experiment is recorded and the performance of the robot has been discussed in Chapter 4.

5.2 Recommendation

To obtain better performance for this robotic hand, the actuation system needs to be improved such as used high torque servo. For further development, the actuating system can be made for every finger joint where every joint can be controlled independently. This type of actuating system will covers 3 DOF for every finger flexion and produce large grasping area. The grasping ability can be improved more by designing the robotic hand with more Degree of Freedom (DOF) that supports all 3 axes which are X-axis, Y-axis and Z-axis of the finger movement. This design will help the robotic hand grasped different shape of objects such as sphere, rectangle, cylinder and so on.

In term of electrical system, it can be improved by develop new power source that have constant power output. This problem in this project is the battery drain too fast especially on the robot side. It is because the battery required supplying power to Arduino Xbee and also the servo. The main component that affects the power consumption is the Xbee module because high power is require in order to performing a reliable and stable communication.

REFERENCES

[1] M.K.Ikhwan, "Multi Fingered Robot Hand Robot Operates Using Tele-Operation", Master of Electrical Engineering, Universiti Tun Hussein Onn Malaysia, 2011.

[2] C.S.Fahn and H.Sun, "Development of a Data Glove with Reducing Sensors Based On Magnetic Induction", *in IEEE Transactions On Industrial Electronics*, 2005.

[3] M.Atif, "Multi Fingered Robot Hand In Industrial Robot Application Using Tele-Operation", thesis Degree of Master of Electrical Engineering, University Tun Hussein Onn Malaysia, 2012.

[4] V.Srovnal and Z.Machacek, "Wireless Communication for Mobile Robotics and Industrial Embedded Devices", *in Eighth International Conference on Network*, 2009.

[5] A.M.M.Ali et al, "Development of Artificial Hand Gripper by using Microcontroller", *in International Journal of Integrated Engineering*, 2011

[6] I.Gaiser et al, "A New Anthropomorphic Robotic Hand", *in IEEE-RAS International Conference on Humanoid Robot*, 2008.

[7] R.Ozawa et al, "Supervisory Control of a Multi-Fingered Robotic Hand System with Data Glove", *proceeding of the IEEE/RSJ International Conference on Intelligent Robots and System*, 2007.

[8] A.H.Ismail et al, "Investigation of Homogeneous Multi Robots Communication via Bluetooth", *in IEEEl Symposium on Computer Applications and Industrial Electronics*, 2012

[9] R.Coiva, R.Haschke, H.Ritter, "Development of An Intelligent Object for Grasp And Manipulation Research", *in the 15th International Conference on Advanced Robotics*, 2011

[10] M.T.Rashid, A.A.Ali, R.S.Ali, "Wireless Underwater Mobile Robot System Based On Zigbee", *in IEEE International Conference on Future Communication Networks*, 2012.

[11] K.H.Lin, H.S.Lee and W.T.Chen, "Implementation of Obstacle Avoidance and Zigbee Control Functions for Omni Directional Mobile Robot", *in IEEE International Conference on Advanced Robotics and its Social Impacts*, 2008.

[12] S.Nishino et al, "Development of Robot Hand with Pneumatic Actuator and Construct Of Master-Slave System", *Proceedings of the 29th Annual International Conference of the IEEE EMBS Cite Internationale*, 2007.

[13] I.Yamano and T.Maeno, "Five-Fingered Robot Hand Using Ultrasonic Motors and Elastic Elements", *Proceedings of the IEEE International Conference on Robotics and Automation*, 2005.

[14] J.W. Lee and T.W.Kim, "Design and Experimental Analysis of Embedded Servo Motor Driver for Robot Finger Joints", *in IEEE International Conference on Ubiquitous Robots and Ambient Intelligence*, 2011.

[15] Y.Hasegawa, M.Shikida, D.Ogura and K.Sato, "Glove Type of Wearable Tactile Sensor Produced By Artificial Hollow Fiber", *in IEEE International Conference on Solid-State Sensors, Actuators and Microsystem*, 2007.

[16] N.Tongrod, S.Lokavee and T.Kerdcharoen, "Gestural System Based On Multi-Functional Sensors and Zigbee Networks for Squad Communication", *in IEEE International Conference on Defense Science Research*, 2011.

[17] G.Saggio, "Bend Sensor Arrays for Hand Movement Tracking In Biomedical Systems", *in IEEE International Workshop on Advances in Sensors and Interfaces (IWASI)*, 2011.

[18] H.Du, W.Xiong, Z.Wang and L.Chen, "Design of a New Type of Pneumatic Force Feedback Data Glove", *in IEEE International Conference on Fluid Power and Mechatronic (FPM)*, 2011.

APPENDICES

APPENDIX A

Arduino Coding for Robot Control (Transmitter)

```
void setup()
{ Serial.begin(9600); }

void loop()
{
  // set input pin
  int sensor1 = analogRead(A1);
  int sensor2 = analogRead(A2);
  int sensor3 = analogRead(A3);
  int sensor4 = analogRead(A4);
  int sensor5 = analogRead(A5);

  //mapping sensor value
  int value1 = map(sensor1, 515,  80, 0, 179); // (7)
  int value2 = map(sensor2, 500,  50, 0, 179); // (8) index
  int value3 = map(sensor3, 520, 140, 0, 179); // (9)
  int value4 = map(sensor4, 350, 620, 0, 179); //(10)
  int value5 = map(sensor5, 260, 530, 0, 179); //( 11)
 delay(50);
      Serial.write(value1);
      Serial.write(value2);
      Serial.write(value3);
      Serial.write(value4);
      Serial.write(value5);
}
```

APPENDIX B

Arduino coding for robot control (receiver)

```
#include <Servo.h>
Servo myservo1;
Servo myservo2;
Servo myservo3;
Servo myservo4;
Servo myservo5;
Servo myservo6;

void setup()
{
  //set servo pin
  Serial.begin(9600);
  myservo1.attach(2);
  myservo2.attach(3);
  myservo3.attach(4); // index
  myservo4.attach(5);
  myservo5.attach(6);
  myservo6.attach(7);
}
void loop()
{
  if (Serial.available() >= 5)
  {
    //read serial data from input pin
    byte temp1 = Serial.read();
    byte temp2 = Serial.read();
    byte temp3 = Serial.read();
    byte temp4 = Serial.read();
    byte temp5 = Serial.read();

    Serial.print(temp1);
    Serial.print("\t");
    Serial.print(temp2);
    Serial.print("\t");
    Serial.print(temp3);
    Serial.print("\t");
    Serial.print(temp4);
    Serial.print("\t");
    Serial.println(temp5);

    myservo1.write(temp1); //thumb1
    myservo2.write(temp1); //thumb2
    myservo3.write(temp2);
    myservo4.write(temp3);
    myservo5.write(temp4);
    myservo6.write(temp5);
  }
}
```

APPENDIX C

Accelerometer coding (x-axis)

```
void setup()
{  Serial.begin(9600); }
int analog_x; float vol_x; float add_x; float g_x; float degree_x;

void loop()
{
  analog_x=analogRead(0);
  vol_x=analog_x*5.0/1024; //convert analog_x-->voltage value(v)

  add_x=vol_x-1.72; //calculate the added x axis voltage value
  g_x=add_x/0.78; //calculate the gram value

  if(g_x<=1&&g_x>=-1) //to prevent the overflow of asin(x).( If x>1 or x<-1, asin(x)=0)
  { degree_x=asin(g_x)*180.0/PI; //calculate the degree value  }
  //fix the overflow condition
  if(g_x>1)
  degree_x=90;
  if(g_x<-1)
  degree_x=-90;

  Serial.print("x:");
  Serial.println(degree_x);
  delay(200);

}
```

APPENDIX D

Accelerometer coding (y-axis)

```
void setup()
{  Serial.begin(9600); }
int analog_y; float vol_y; float add_y; float g_y; float degree_y;

void loop()
{
  analog_y=analogRead(1);
  vol_y=analog_y*5.0/1024; //convert analog_y-->voltage value(v)

  add_y=vol_x-1.74; //calculate the added y axis voltage value
  g_y=add_x/0.78; //calculate the gram value

  if(g_y<=1&&g_y>=-1) //to prevent the overflow of asin(y).( If y>1 or x<-1, asin(y)=0)
  { degree_y=asin(g_y)*180.0/PI; //calculate the degree value  }
  //fix the overflow condition
  if(g_y>1)
  degree_y=90;
  if(g_y<-1)
  degree_y=-90;

  Serial.print("y:");
  Serial.println(degree_y);
  delay(200);

}
```

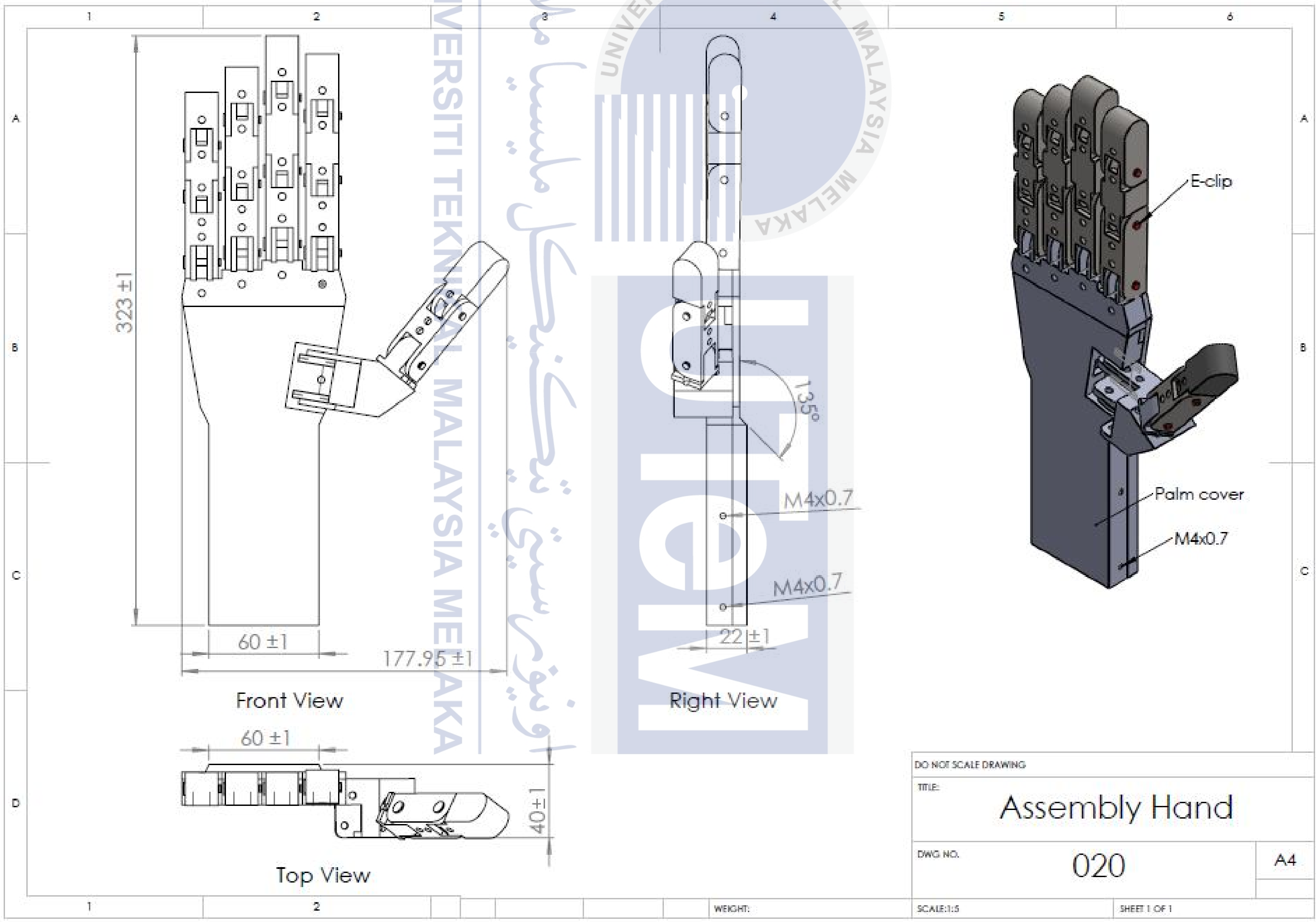
APPENDIX E

Arduino coding to read sensor output (synchronize with Bridge Control Panel)

```
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

// the loop routine runs over and over again forever:

void loop()
{
  // read the input on analog pin 0:
   int data = analogRead(A0);

   Serial.write(data>>8);
   Serial.write(data&0xff);
}
```

APPENDIX F

Hand design



APPENDIX G

Project planning

PROJECT PLANNING										
	2013				2014					
Project activities	**September**	**October**	**November**	**December**	**January**	**February**	**March**	**April**	**May**	**June**
Research and study on topic										
Circuit design										
Hardware design										
FYP 1 presentation										
Hardware development										
Software development										
Experiment and collect data										
Troubleshooting										
Hardware performance analysis										
Final presentation										

APPENDIX H

K-Chart